電腦時代

COMPUTING AGE

4

微型電腦用家雜誌

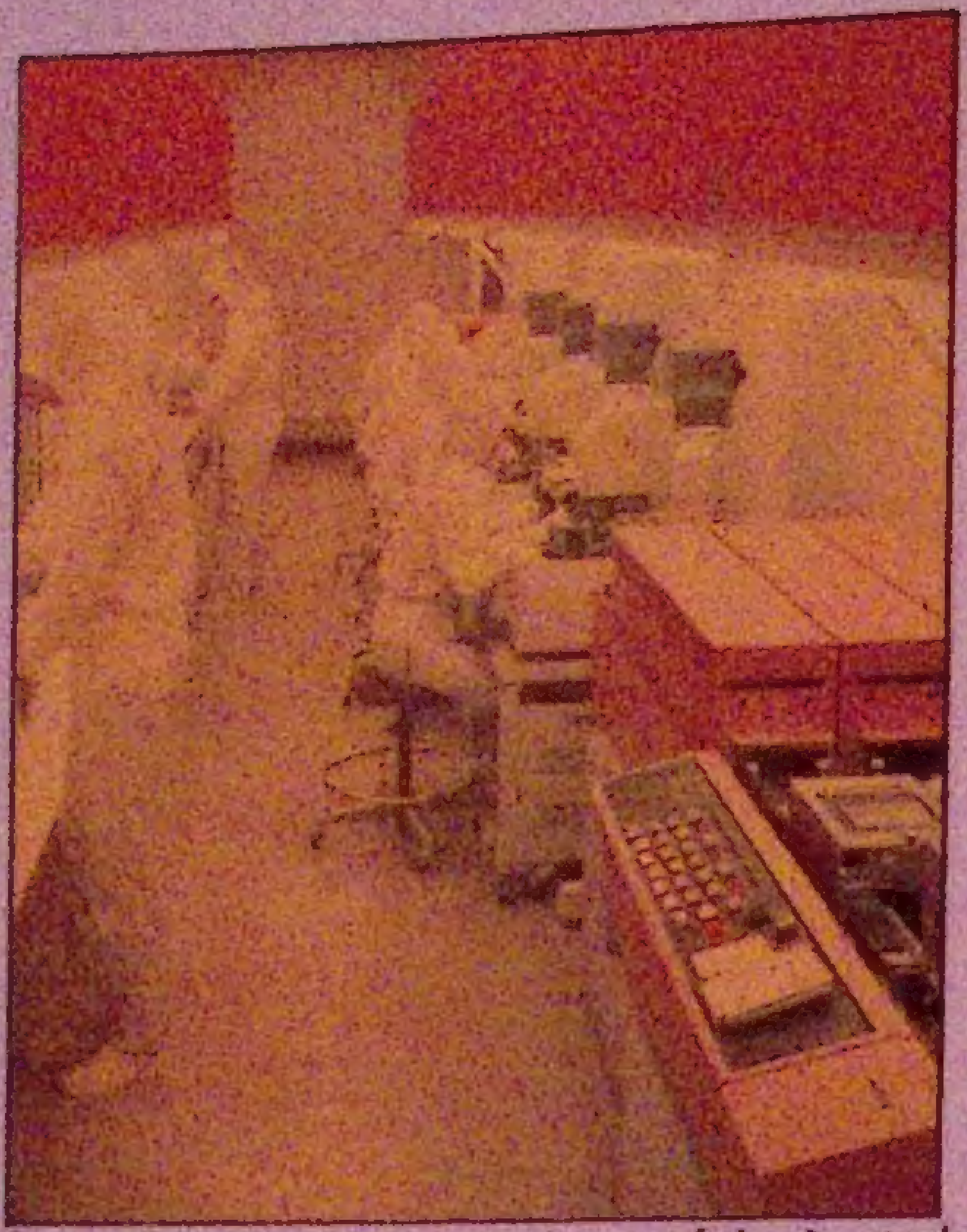
HK$8.00

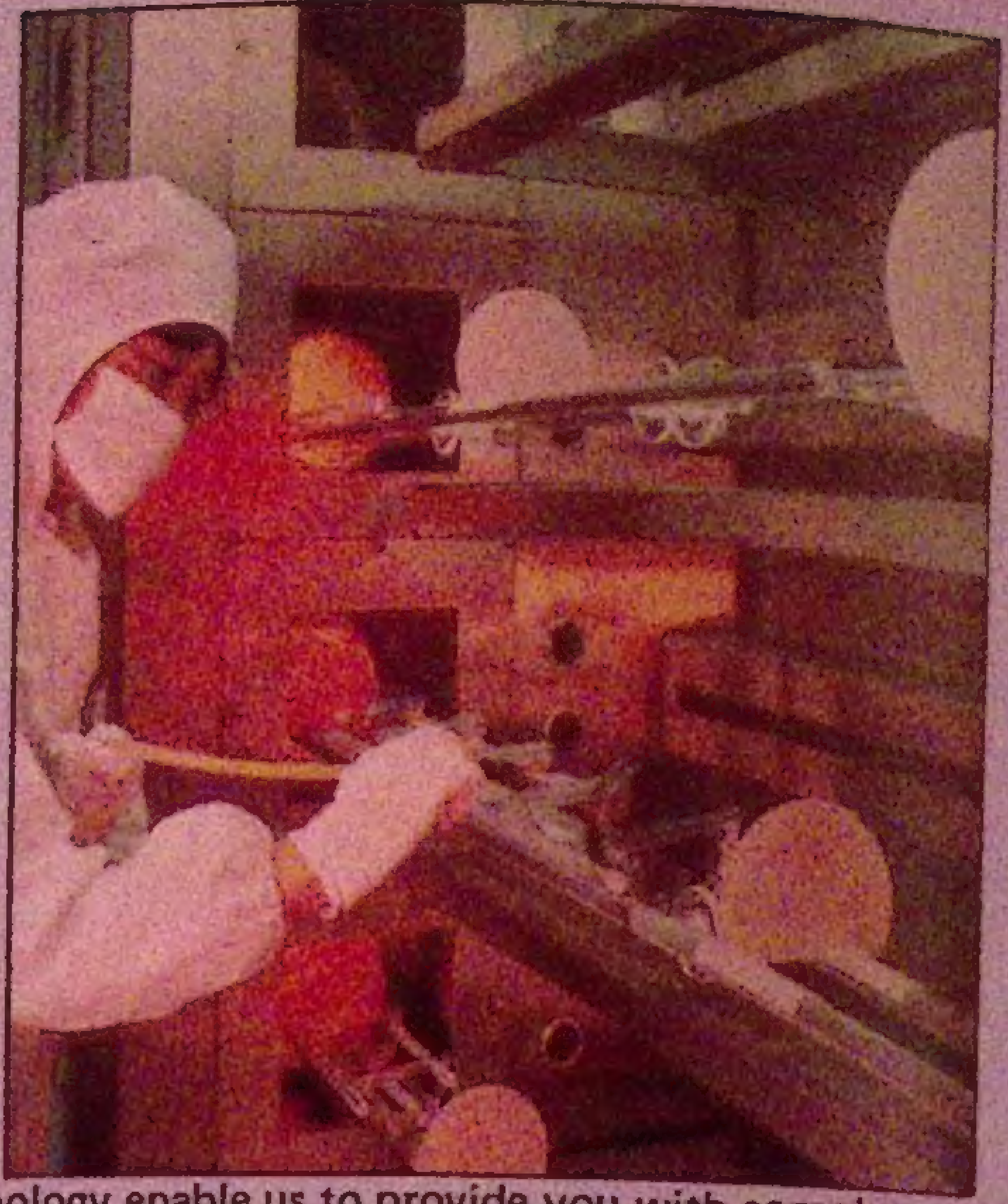
- 揭開蘋果最新操作系統 PRODOS 的神秘面紗
- 六代同堂——APII微電腦之進化史
- CP/M系統的闡釋及其應用（連載篇）
- 被人誤解的LOGO 語言
- 電腦動畫圖案製作原理





Our latest CAD facilities are just one indication of our strong commitment to customer design services.





State-of-the-Art facilities and state-of-the-Art technology enable us to provide you with complete silicon-foundry services. Turn-around cycle is just 20 DAYS.

From A New Leader in Silicon Gate CMOS Technology

Our success has been based on manufacturing circuits on the complimentary MOS (CMOS) technology featuring consistently high performance and reliability. Because of the intrinsic superiority in respect to noise immunity, a wide variety of voltage — supply conditions, thrifty use of power, and temperature insensitivity, the CMOS technology is especially ideal for use in harsh environments where no other technology can withstand.
We carry a standard IC line and regularly provide customer design and foundry services. The turn-around cycle for foundry services can be as short as 20 days.

Product Number	Process	Description	TTL or CMOS Compatible	Pin Compatible with
MICROPROCESSOR, MEMORY AND LOGIC				
HKE65SC02	CMOS	8 bit Microprocessor	TTL and CMOS	NMOS 6502
HKE5114	CMOS	1Kx4 Static RAM	TTL and CMOS	2114 4K SRAM
HKE74HCTxxx	CMOS	High Speed CMOS Logic Family	TTL and CMOS	74 Series TTL
HKE74HCxx	CMOS	High Speed CMOS Logic Family	CMOS	74 Series TTL
WATCH CIRCUITS				
HKE151	CMOS	32 KHZ analog	Watch circuit	MB 151
HKE5001	CMOS	3½ digit 5 function multiplexed LCD		MSM 5001

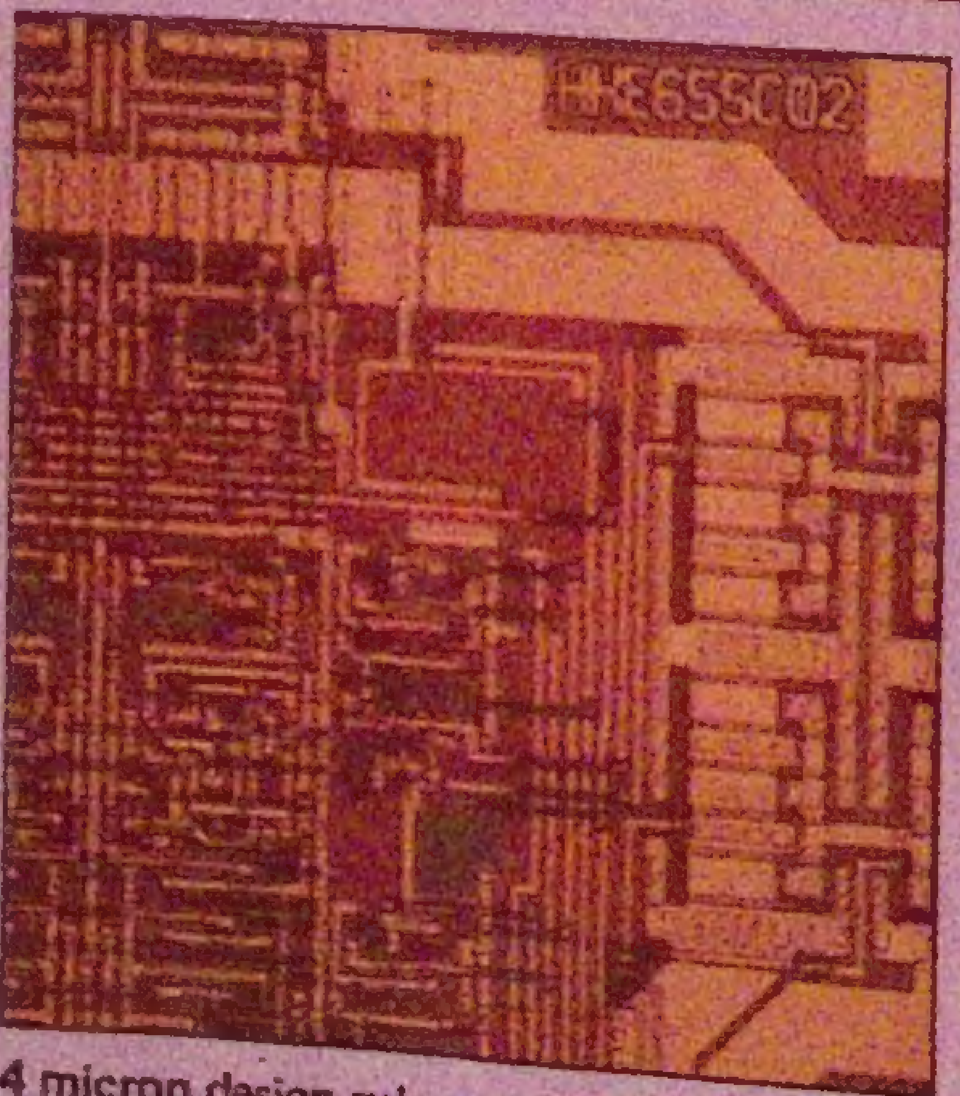
THE SYSTEM DIVISION OF HUA KO
is developing and marketing personal computers and medium size computers for general purpose and turn-key applications.

THE ORGANIC MICRO-modular units for expanding and upgrading Sinclair ZX81 and ZX Spectrum personal computers.

- PERSONA — Interface module enables Organic Mirco to grow on your computer
- MINIMAP — Memory mapping to extend to a maximum of 1M bytes!
- RAM 08 — Low cost memory expandable from 2K to 8K
- RAM 16 — 16K add-on memory at remarkable low cost
- RAM 64 — A TRUE 64K add-on memory
- DROM — For non-volatile storage of programs and data
- TOOLKIT — A module to take 8K bytes of utilities in EPROM/ROM
- PERICON a — Providing 24 lines of input/ output
- PERICON b — 24 lines of heavy duty output-access the outside world
- PERICON c — Drive an 80 column printer with Centronics type interface
- SONUS — Three-voice synthesizer, sound generator with stereo option!
- LINK a — An 8 channels A/D Converter

IBM PC compatible Add-on Memory board
- HKE 64-pc — 16K Bytes
- HKE 128-pc — 128K Bytes
- HKE 192-pc — 192K Bytes
- HKE 256-pc — 256K Bytes

IBM is the registered trademark of International Business Machines.



4 micron design rule and ISO-CMOS silicon gate technology are used in the manufacture of the HKE65SC02.



The very first product announced by HUA KO is an 8 bit CMOS microprocessor, the HKE65SC02, pin-to-pin plug-in compatible with the NMOS 6502.


DISK:
4
DISK:

RITEMAN 超級印字機

超值售價　超速印字　超薄機身

特點：

- 機身體積爲普通印字機一半。
- 超速印字，每秒可印120字。
- 雙向式穿梭印字，每分鐘可達63行。
- 多種印字功能，可印四國文字，更可作Graphic Dumping（需另加Graphic Interface）。
- 採用新設計之盒式絲質墨帶，用戶可補充油墨重覆使用，操作簡易，墨帶壽命高達一百萬字。
- 減少更換墨帶次數，不斷節省金錢，物超所值。

長期節省開支，半年慳到差價，二十個月慳到一部新機。

	每盒售價	可印字數	每年印字數	需用墨帶	總開支
普 通 墨 帶	HK$60	600, 000字	20, 000, 000字	33盒	HK$1, 980
Riteman可補充油墨墨帶	HK$40	1, 000, 000字	20, 000, 000字	20盒	HK$ 800

每年節省開支達HK$1, 180.

獲美國聯邦通訊委員會，標準檢定合格，証書編號 CWT9Z6RITEMAN，以保証品質。

享有代理商保証，才是專業性保用，慎防水貨及二手貨，請索取全新保用証，如証上有任何塗改，即自動作廢。如有疑問，請立即向本公司



特約經銷商：

Golden Micro Computer Co.
3-793168, 3-602564

Profit's Computer Co.
3-780983

Sunray Electronics Co.
3-7237390

總代理
VINCENT COMPUTER CENTRE LTD.
(A Subsidiary of Vincent Group)
雲生電腦有限公司
（雲生集團附屬機構）

九龍尖沙咀柯士甸路22-26號好兆年行608室　電話：3-7211288-0
電訊：HX38316 VCC HK

蘋果公司的一次訪問

A HEART TALK WITH APPLE

本刊記者



美國蘋果電腦公司遠東區商務經理艾思復（右）及洲際營業部董事莊嘉樂（左）接受本刊的訪問。

香港電器市場潛力之大，是衆所週知的，尤以電腦爲甚，所以最近美國蘋果電腦公司也爲其新型號——MACI—NTOSH電腦的面世，派出多位高級職員越洋來港作推廣。本雜誌藉此機會訪問了該幾位職員。

在接受本刊訪問時，蘋果公司的洲際營業管事莊嘉樂 JOHN N. GAL—LO 及遠東業務經理艾思復 SAMI ASFOUR 一致認爲香港是遠東區最具潛質的市場，其次是新加坡、馬來西亞等地，故此今年將集中精力在港大展拳脚。

對於未到手的肥肉——中國大陸，他們也滿懷信心，艾思復表示，他曾兩度訪問北京，查看當地市場。事實上，現時部份的機關已購入蘋果電腦，而且當局又計劃擴充電腦教育。最令艾氏高興的，就是他在北京街頭與學機械的內地人，提起「蘋果牌」電腦時，他們也對這個名牌有所聽聞又認識。

爲了滿足華人社會的需要，新的 MACINTOSH 將於今年底推出中文版本，這不但爲他日大陸市場作舖路，而是電腦本地化一大壯舉。負責電腦本地化工作的莊嘉樂指出，中文是各國文字中最困難的一種，因爲它是方塊字，不似其他用字母拼音，所以公司在做妥法文、意大利文、比利時文、荷蘭文、瑞士、瑞典、拉丁、日文、葡萄牙、以色列等文字版本的 MACINTOSH 後，才推出中文版本。

在芸芸衆中的電腦產品，劇烈的競爭下，新機要打入市場，實在極之困難，MACINTOSH 又怎樣面對勁敵呢？

擁有二十年國際市場經驗的莊嘉樂輕鬆的說：「現時的新機一定要具備三個基本條件，才能在衆多的產品中，一枝獨秀。一是出產公司有良好的聲譽；二是有充足的軟件，及獨立製件商在背後支持與發展；三是容易使用，性能強。這些 MACINTOSH 都擁有。」

首先美國蘋果電腦公司是電腦行業的表表者，他的產品均能適合大小規模的公司，幼稚園至大學生都選用。八三年十二月三十日至八四年首季業績營業額高達三億一千六百二十萬美元，比上年同期之二億一千四百三十萬美元上升百分之四十八。淨收益額爲五百八十萬。最近美國大學剛簽下五千萬元的合約，即廿五間著名大學（包括哈佛、耶魯等）在未來數年內會採用蘋果產品，爲電腦教育邁進一步，這可見蘋果牌產品的可信賴發度。至於本港市場推廣上，蘋果公司最近與太平洋行合作，如此強大的實力，對勁敵又何有懼哉呢？

在軟件支緩上，目前的有逾百家公司爲 MACINTOSH 發展軟件及硬件，最流行的 LOTUS 1—2—3 綜合商用軟件也即將加入此行列。預料適合 MACINTOSH 電腦應用的軟件將有九成是由獨立軟件商供應。估計到今年底，將有超過五百種軟件系統編寫完成，爲 MACINTOSH 及新版本的L—ISA 電腦使用。

現今社會，事事都講求效率，所以 MACINTOSH 利用「老鼠仔」來按圖索驥，簡單方便，只須練習數小時便能操作自如。

切勿少覷這個面積十吋乘十吋，僅重十七磅的電腦，它是以卅二位元的 MOTOROLA 68000 微處理器爲根基，內置記憶體具有 192 K BYTE 記憶容量，其中 ROM佔 64 KBYTE，而 RAM記憶則爲 128 K BYTE。內置多音度聲音產生器，可發出高質素人類語言及音樂，又有高速 RS—232 及 RS—422 APPLEBUS及串列挿孔。

也許，你以爲如此超級微型電腦的售價，定是價值不菲的。但事實上，你只要付出二千四百九十五美元，便可擁有這部配合時代節拍的新產品。

由於上述種種有利的條件，蘋果公司在加州費里蒙斥資二千萬美元，開設一所高度自動化工廠，專門生產 MA—CINTOSH。製造一部電腦，僅需時廿七秒！

版權，膺品，是名牌電腦最煩腦的事，尤以東南亞的問題最大，蘋果公司日後會採取何種對策呢？

莊嘉樂指出：「最近公司在美國，與台灣打了多場官司，大有成效，就以與 FRANKLIN 電腦的訴訟爲例，對方同意付予本公司二百五十萬美元，並答應不再侵犯蘋果版權。」

「這些法律行動很快就會在香港展開。其實，現時香港、台灣及新加坡等地各政府對膺品亦積極採取行動，以淸除舊有的膺品市場的印象。」

莊氏最後說：「新出的蘋果產品，已落了很多保護手脚，當然不能說絕難抄襲，但如果冒險行事，其後果實是堪虞！」■

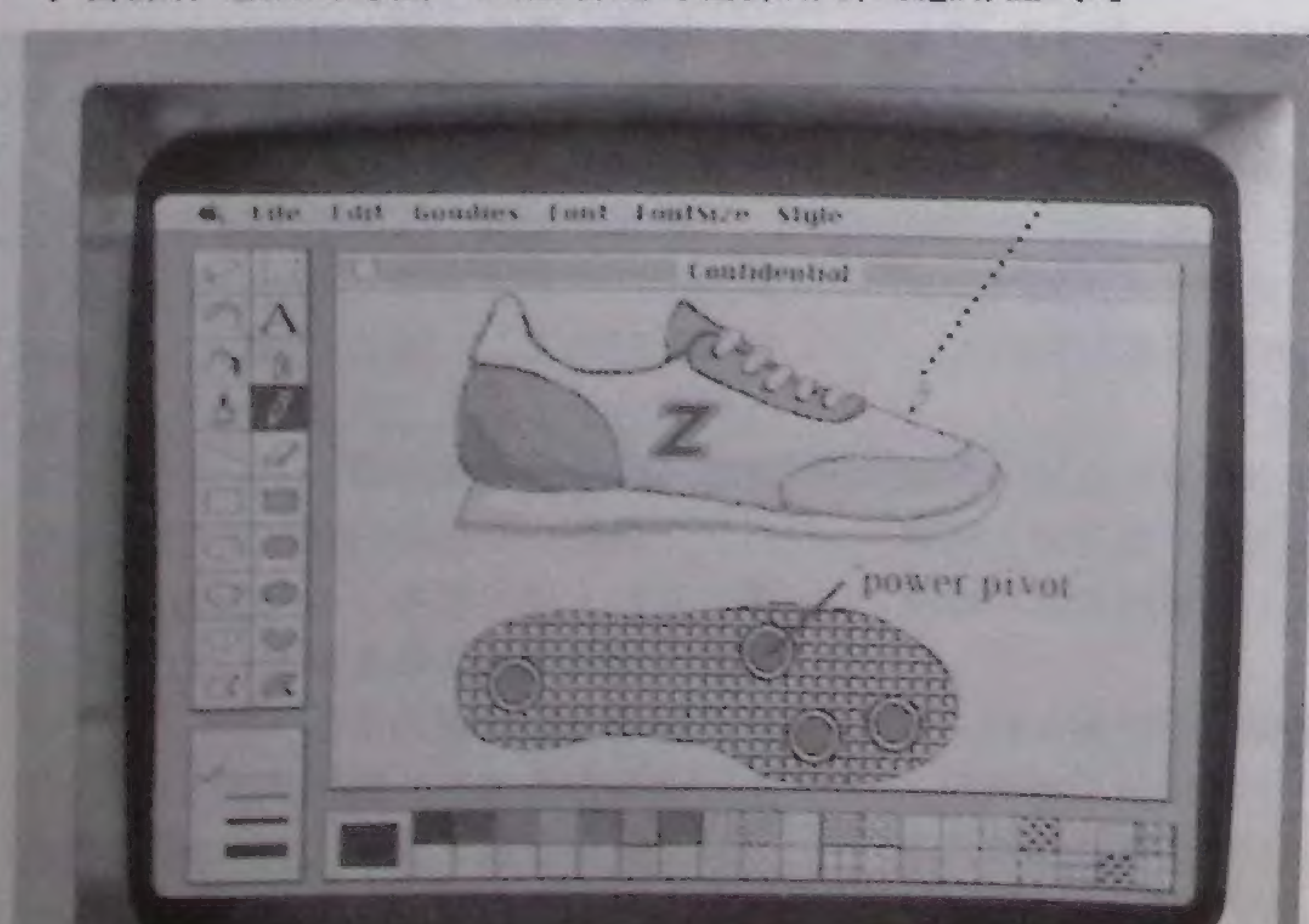
蘋果MACINTOSH 新電腦在港正式面世



盛傳已久的蘋果 MACINTOSH 電腦，上月（一月廿八日）在港正式宣告面世。這是一款易於使用的新型私人電腦系統，適合工商界，專業人士及學生作一般廣泛用途。

這種新型電腦的操作方法極之簡便，使用者只需移動一個指示電腦操作的裝置——「老鼠仔」，便可根據螢幕上所顯示各種功能的圖案作出選擇，指示電腦進行有關工作。例如以文件夾代表提存檔案資料，計數機代表運算計數，以及記事簿代表紀錄約會及工作備忘，而垃圾箱圖形則代表刪棄，用戶只須用「老鼠仔」操作游標，指向其中一個圖案，便可達到所需功能目的，同時每種應用軟件均可交互操作。

由於大多數人都無法或不願花費多達二十至四十小時來學習操作電腦的方法，更甚者是每當採用新的應用程式時，



應用軟件中所有指令，皆採用圖形代表，一看便懂。

又要多花費十個小時來學習使用。但 MACINTOSH 電腦則剛好相反，因爲它運用老鼠仔「按圖索驥」，指揮電腦操作簡單方便，只須練習幾小時便能操作自如。

MACINTOSH 的唯讀記憶體乃採用麗莎電腦技術製成，設有電腦操功能表，視窗及各式圖表功能。

這具輕巧精密的 MACINTOSH 電腦主機放在一般辦公室桌上使用，佔用面積不過十吋乘十吋，重量僅爲十七磅，較一般市面最暢銷的便携式電腦更要輕盈。

若需要大量貯存資料，這類新式電腦的三吋半磁碟機及三吋半磁碟便可大派用場。這種小型磁碟體積雖小，但每面可貯存資料多至 400 K BYTE（大約相當於一百頁普通文件紙張的容量），磁碟並附有堅固的塑膠套，以防刮損。

MACINTOSH 電腦的設計主旨要進軍整個世界市場，因此在設計上避免使用英文操作，甚至機身上的符號也不用英文，鍵盤上的各種功能鍵、挿孔及維修指示以圖案顯示。至於電腦的唯讀記憶器亦不便使用英文符碼。同時它的軟件採用革新建構，方便電腦翻譯器改編軟件，適用任何一種語言。

現時在美國推出的是英語版本。蘋果公司計劃每個國家，都有使用該國語言的 MACINTOSH 版本。

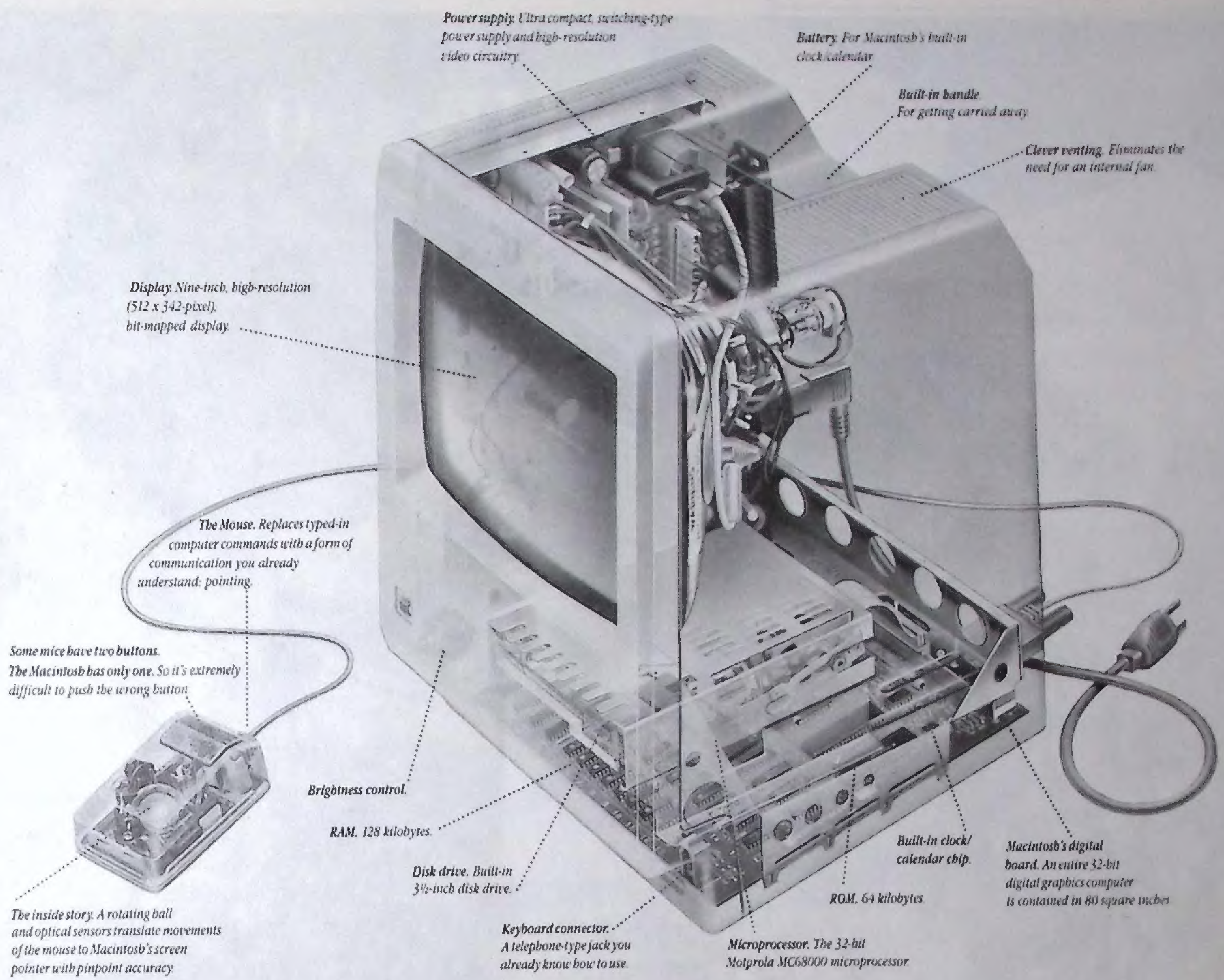
蘋果公司將按下列時間表在世界各地推出 MACINTOSH 電腦：

一九八四年四月：英國、愛爾蘭、法國、西德、意大利及澳洲。

一九八四年七月：荷蘭、瑞士、瑞典、拉丁美洲國家及日本。

一九八四年十月：西班牙、葡萄牙、以色列、中東及北歐三國。

同時，MACINTOSH 亦會有中文版本。不過目前由於有關中文在輸入方法尙在研究中，故未能有準確的時間 ▣



MACINTOSH 電腦的特徵如下：

主機：

●附設內置式螢幕顯示屏及三吋半磁碟機，連同分體式鍵盤共重十七磅。

●採用 MOTOROLA 68000 32 BIT 微處理器，運作頻率爲 8MHZ。

●內置 192 K BYTE 記憶，其中 64 K爲 ROM，128 K爲 RAM。

●內置多音度聲音產生器，可複製高質人類語言及音樂。

●內置式高速 RS－232及RS－422 APPLEBUS 及串行通訊器插孔。

●老鼠仔及外置磁碟機連接器。

●可支援點形繪圖軟件。

●電池供內置完時時鐘及日曆使用。

老鼠仔：

●一個滾動外殼、附有按鈕，用電綫插進主機背後之老鼠仔連接器。其小巧輕盈如掌握，只須放在枱面上移動，便可操縱螢幕上的游標隨着轉動。

●鍵盤及監察器

分體式鍵盤，設有五十七鍵（另有十八鍵的小鍵盤另供選用）。

●九吋畫面黑白螢幕顯示器，高解像力，512×342 點磁碟儲存。

●三吋半磁碟機。400 K BYTE。

●三吋半外置磁碟機（選用配件）。

●大容量 400 K BYTE三吋半外置碟磅機。

●蘋果影像印字機（多種字體、圖像、比例對稱文本、混合圖文，可印普通紙及電腦紙）。

●蘋果調制解調器及 MACTERMINAL軟件（自動攪號碼及自動答話，有 300 BAUD及300／1200 BAUD款式。

●蘋果會計師式數碼鍵盤。

● MACINTOSH 軟質手挽袋。

●APPLELINE容許 MACINTOSH 仿效 IBM 3277 及 3278 型主機電腦操作。

●附有防盜鎖，可將MACINTOSH 電腦及鍵盤鎖在寫字枱或普通枱子上。

● MACWRITE

文字處理軟件，有多種字體大小，搜索及修改功能，並可從其他程式中提取圖文，合併在公函或報表上。

● MACPAINT

繪圖說明程式，可代替畫筆、鉛筆和擦膠，並有大量圖形和畫像，可供繪畫各式圖案之用。

● MACDRAW

供作創製各種曲綫圖，組織圖及流程圖用。

● MACPROJECT

供作創製流程及資源應用圖表之用。

● MACTERMINAL

容許 MACINTOSH 仿效 DEC VT 100，UT 52，TTY， 而配上 APPLELINE 後，更可仿效 IBM 3277 及3278終端機。

●MACINTOSH PASCAL，MACINTOSH ASS—EMBLER／DEBUGGER，MACINTOSH LOGO BASIC。■

奇妙的 HP 150

指尖底下的電腦・輕觸螢幕自動操作



微型電腦與用家之間的聯絡溝通繼光筆、「老鼠仔」之後，又有一項重大的發展——利用指尖觸按監視器螢幕來控制電腦工作。

美國 HP（HEWLETT－PACKARD）公司最近推出一部突破性設計的私人商用電腦 HP 150 就是擁有一個觸按式感應螢幕。這是全球第一部。

「老鼠仔」介面設計在應用上經已方便了許多電腦使用者。但 HP 這部觸幕式電腦更加使用者無論在時間上或實際應用都更節省時間及提高效率。而且對初用電腦的人士或不喜歡打字的高級行政人員更是天大的喜訊。

使用者在使用時，毋須背熟任何的指令或代碼，只要隨畫面上所顯示的指示，用手指輕輕觸動便可完全控制電腦的操作。

HP 150私人商用電腦這項獨有設計主要原理是利用螢幕框緣的光敏感器，搜索指尖在螢幕上所指的座標位置。然後自動執行各項相對的操作。

據設計這部電腦的工程師指出：當初設計 HP 150 時，在輸入媒介方面原意要希望電腦和用家之間，有更直接的溝通。於是曾經嘗試在鍵盤上加上慣常使用的功能鍵掣來配合螢幕上所顯示之指令，不過鑑於在指令改變時，按鍵的功能亦要有相對的轉變，這種情形，就受到一定鍵盤空間限制，相反，利用這項觸感應式設計，問題就迎刃而解。

事實上，觸幕式螢幕並非一種新發明的輸入媒介，它經存在很久。問題只是如何將它的生產成本控制在合理範圍之內。這種技術，只需採用廉價的發光二極管及光電二極管。

HP 150私人商用電腦的體積只佔一本電話部大小的平面空間，但卻內藏不少的功能。例如在頂部的一部熱感式印字機及內置軟或硬（自由選擇）大容量磁碟機。它同時採用 3 ½吋磁碟機。

HP 150系統的零售價大約爲港幣三萬一千元。這套系統包括：鍵盤、採用 INTEL 8088晶片處理器的系統處理機、位變換圖形顯示監示器、256 K BYTE－RAM記憶，並且可以擴展至 640 K。

電腦內部裝置有內置數據通訊埠。其CPU（8088）是以 8MHZ 來運行。目前市面上大部份私人電腦所採用的 CPU速度均爲 5 MHZ之下。

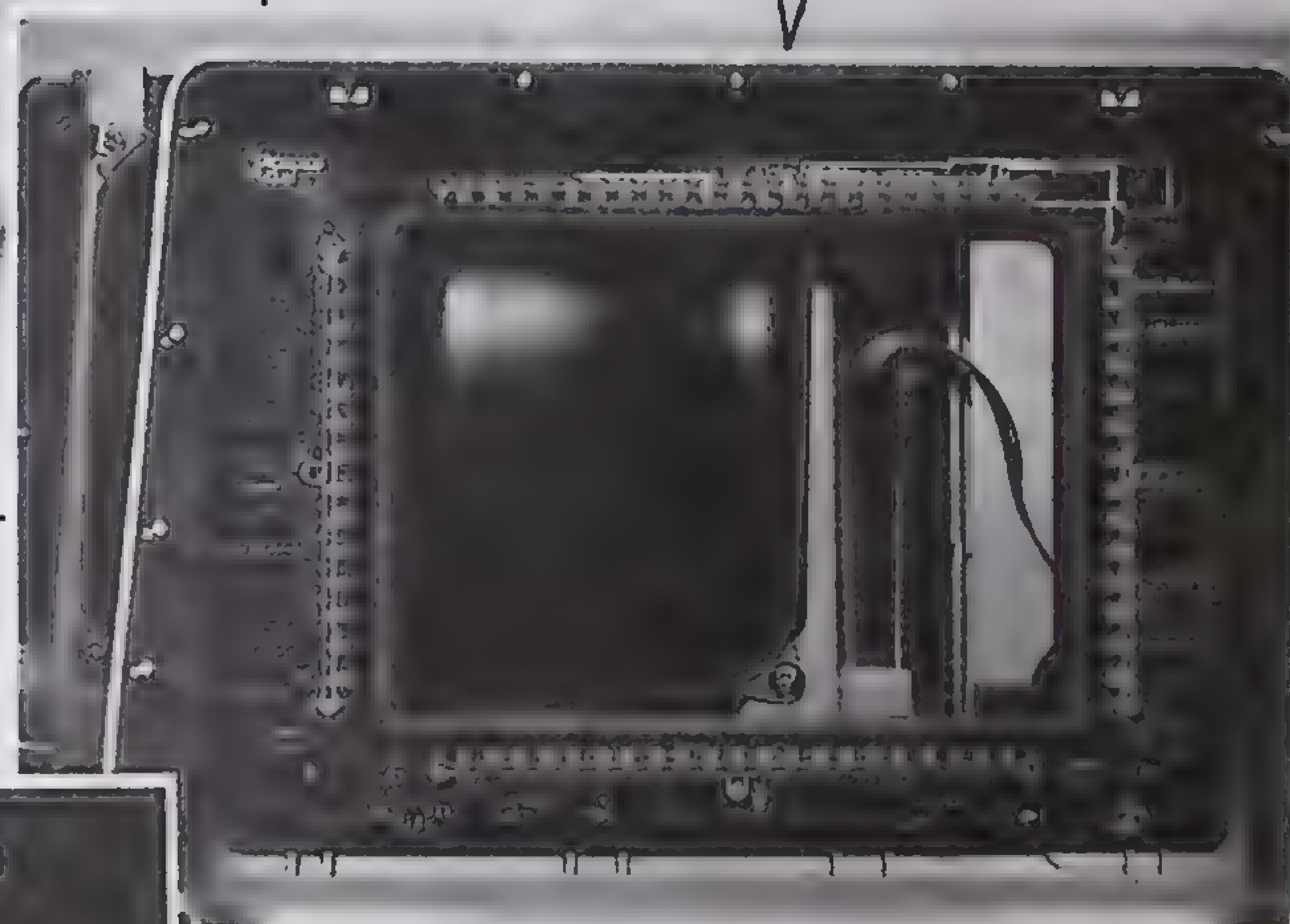
在顯示方面，監視器的性能相當出色，可在9吋顯示屏上（電腦的螢幕）作 720×378 點的字母數字模式顯示，而位訊息圖像顯示卽爲 512×390，比較上來說，較其他電腦顯示佳。螢幕底部留有位置予控制指令顯示，供用家在進行操作時使用。

由於它採用 INTEL 8088爲 CPU，並配有MS DOS 2.0 操作系統，市面上許多通行的應用軟件皆適用於新機之上。這包括：WORDSTAR、VISICALC、PERSONAL CARD FILE、MEMOMAKER等。□

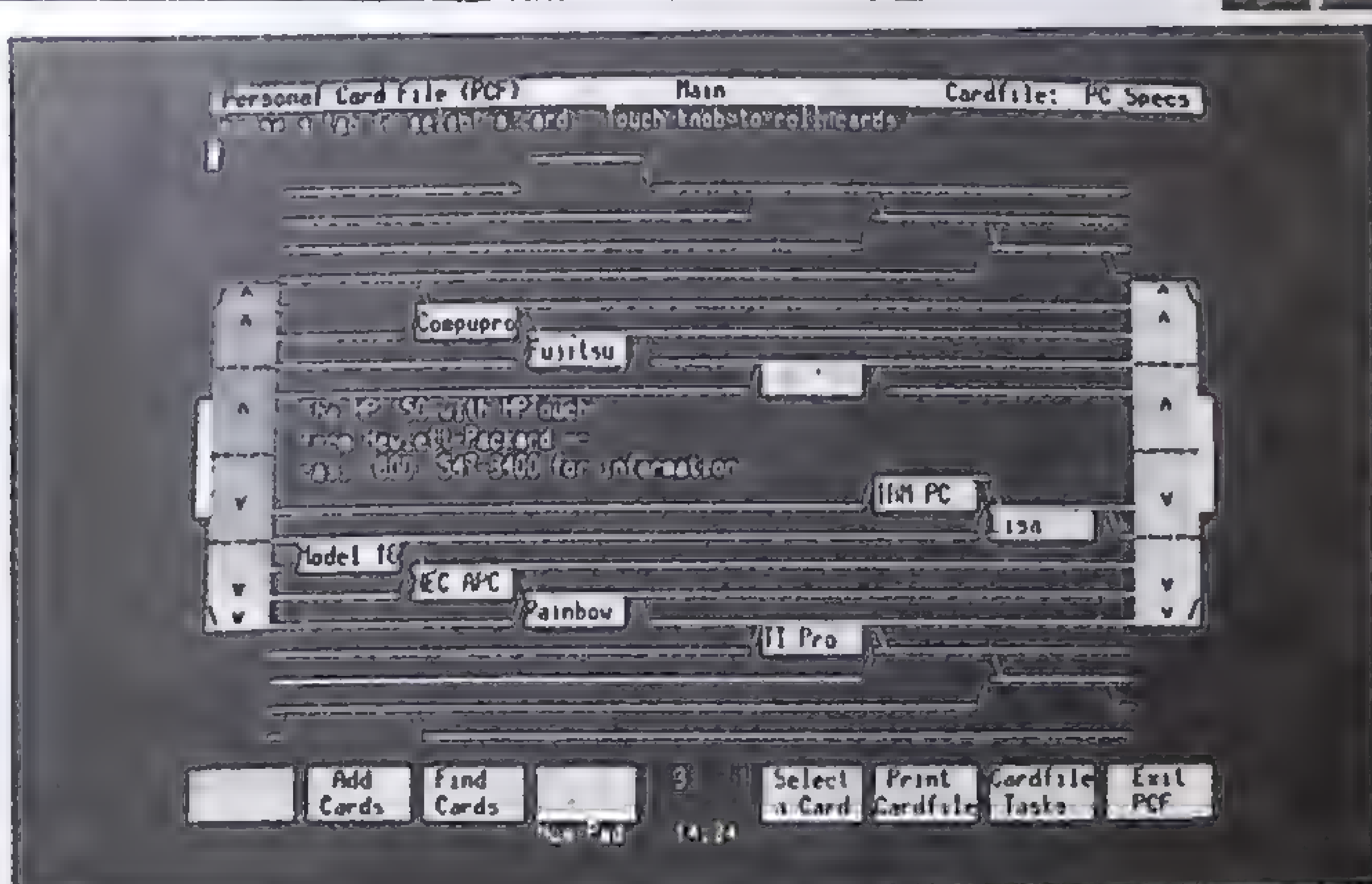


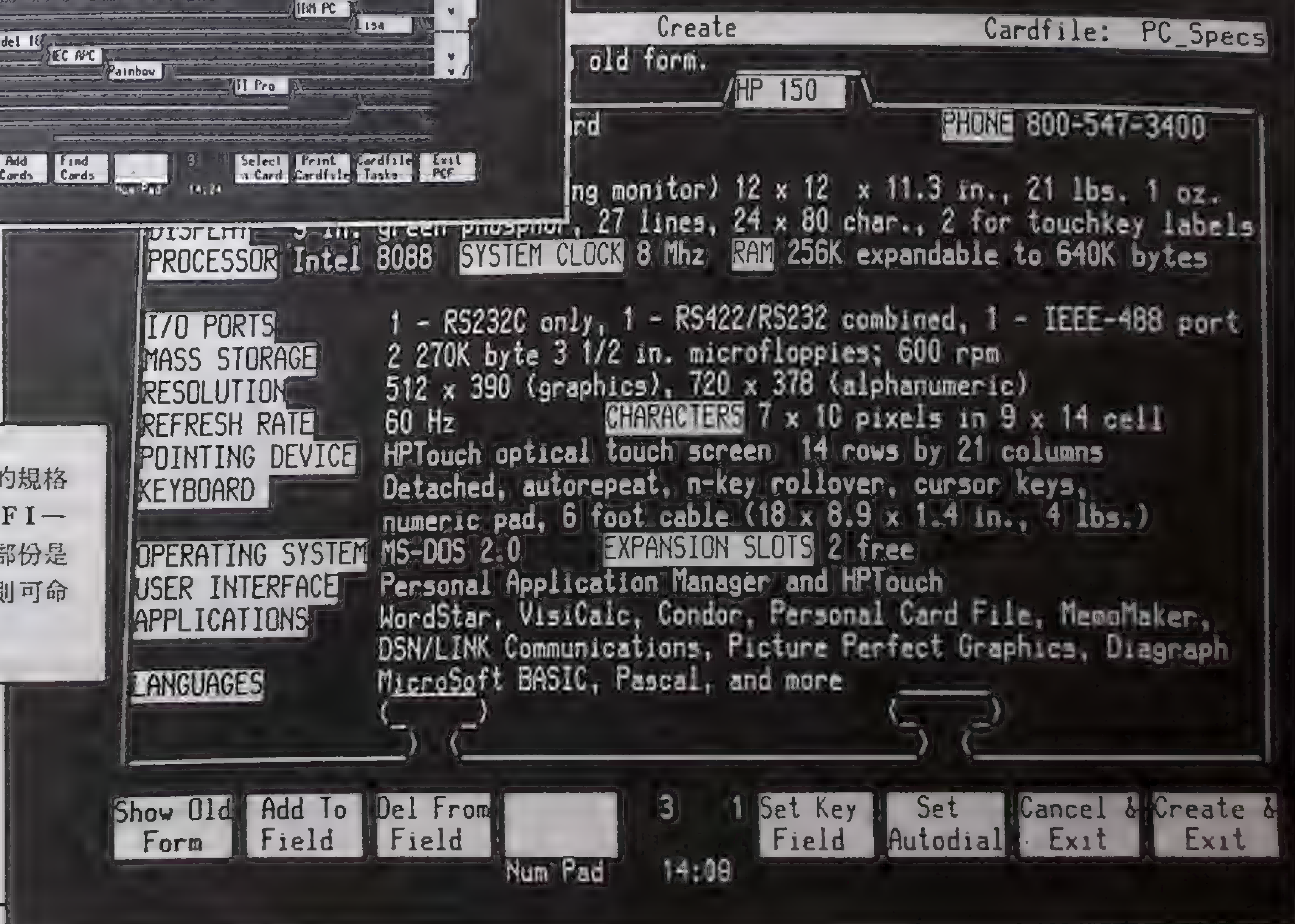
HP 150的背視圖。其頂部是一部內置熱感式印字機。它擁有兩個串行埠（DATACOM 1 及 DATACOM 2）和一個 HP 1 B　總綫（用作接連磁碟機及平行印字機之用）。

HP 150觸幕式設計的秘密——在螢幕邊緣裝有多個光敏感應器及發光二極管用來感覺指尖在螢幕的位置。



HP PERSONAL CARD FILE 軟件。





HP 150私人商用電腦系統的規格，利用 PERSONAL CARD FILE 來顯示給大家看。螢幕最底部份是控制指令，用家只要用指尖輕觸則可命令電腦工作。

將RAM卡作爲磁碟機使用

用RAM卡存取資料比磁碟機快多倍，今期爲大家介紹的這個RAM DISK 程式，就是將RAM卡當作一部磁碟機使用



RAM DISK 是一個機械語言程式，使你可以將16K RAM卡（或 FRANKLIN ACE或 64K APPLE Ⅱe 的頂部16K寄存器）當作磁碟機一般使用。這程式容許你用普通的 DOS 指令（RUN，SAVE，LOAD，OPEN，READ， WRITE，CLOSE）將程式或數據存貯在 R－AM卡上。

這些指令的操作情況沒有甚麼特點，唯一不同的是更快，因爲寄存器的操作比磁碟機快得多。當然，關掉電腦後，寄存器的資料亦會消失，如果你想貯起卡上的資料或程式，可以用 SAVE EMU指令，將 RAM DISK的全部資料（64 SECTOR 的程式或文件）傳送到磁碟上，費時不超過三秒！一隻磁碟可以存起相等於 8 份 RAM卡的記憶。要將任何一份版本由磁碟傳輸回 RAM卡，所費時間亦不超過三秒！

本文前半部主要介紹如何使用 RAM DISK；後半部是討論它如何在 16 K CARD 上使用寄存器。

RAM DISK的使用

使用 RAM DISK 所需要是一部 APPLE Ⅱ 或 D－OS 3.3 的 APPLE Ⅱ PLUS，在 SLOT 0 加插16K RAM卡，APPLE Ⅱe 或 FRANKLIN ACE 亦可以打 0 入程式入電腦後，然後打 BSAVE RAM DISK，A$ 9000，L$2AA，將它貯入磁碟。當有需要時，打入 BRUN RAM DISK 指令，便在兩秒內由磁碟載回電腦，這張 RAM卡就彷似連接到 SLOT 5 的磁碟機一樣了。

只要你在 SAVE 指令後加上"S 5"（即 SLOT 5 的縮寫），便能將一個程式貯起，例如：10 HOME：VT－AB 8＝PRINT "WELCOME TO THE WORL－D OF THE RAM DISK"。

只要打 SAVE WELCOME，S5〔RETURN〕便能將它 SAVE 入 RAM卡內。 RAM DISK 不會理會那部"磁碟機"，例如，你打入 SLOT 5，DRIVE 1或 SLOT 5，DRIVE 2，甚至只是 SLOT 5 也好，其結果都一樣。如果要看 RAM DISK內藏有甚麼 FILE，可以打 CATALOG，S5〔RETURN〕。

剛才每個指令都指明 S 5，但事實上，這並非必要，正如DOS 一樣，RAM DISK 會沿用上一次使用的 SL－OT 和「磁碟機」直至另行通告爲止。

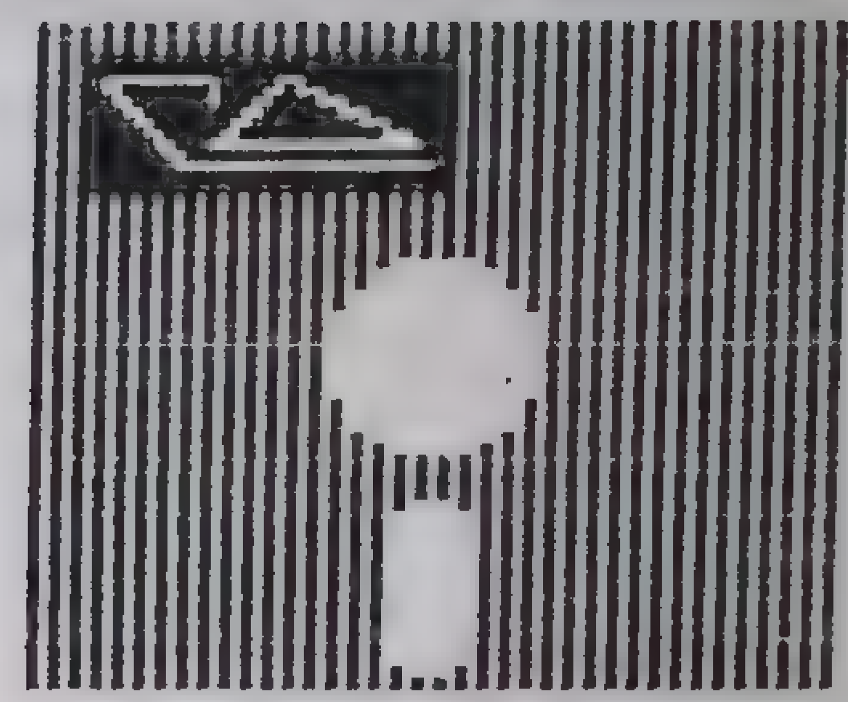
運行以上程式，非常容易，只要鍵入 RUN WELCO－ME〔RETURN〕，便立刻有反應，大致上來說，D－OS 指令仍照常使用，所不同者，是當指令挿有 RAM DISK 的 SLOT 編號時，這些指令會使用RAM DIS－K而不是磁碟機。

一些 DOS 指令，亦能如常操作，例如：10 PRINT CHR $（4）"CATALOG，S5"的作用，和用鍵盤直接打入 CATALOG，S5〔RETURN〕一樣，使螢幕顯示出 RAM DISK 的CATALOG。此外，一些用於文件內貯存數據的指令，如 OPEN，READ，WRITE 和 CL－OSE 等，亦能用於 RAM DISK，只須指明 RAM D－ISK 所處的 SLOT 編號。由於這些指令不能由鍵盤打入操作，所以我們介紹一個簡單的程式，用來顯示傳統磁碟機和 RAM DISK在操作時速度的差別。列表 2 的程式，名爲 RAM DISK TEXTFILE DEMO，會產生兩個 FILES，一個在磁碟，另個 RAM DISK，每個文件包括一個英文字母及一個 DISK 字或 EMU字。當文件產生後，你可經由 RAM DISK 及磁碟機使它恢復在螢幕前，你會發覺，它們的速度相差很大。

特別指令

本程式只有 5 個指令較爲特別，是 RAM DISK 常用

將RAM卡作爲磁碟機使用



的，就是 INIT EMU，SAVE EMU，LOAD EMU，LOCK EMU和UNLOCK EMU。它們所以不同，是因爲它們後面跟著有 EMU。EMU 是一個保留字，因此，你不能用 EMU 作爲任何 BASIC 程式 TEXT FILE 的名稱。

INIT EMU 指令

在 RAM DISK，使用 INIT EMU 有兩個不同之處：第一，如果你用 INIT HELLO 來 INITIAL—IZE 磁碟，磁碟會被 INITIALIZED，但 DOS 不會貯在磁碟上。道理很簡單。RAM DISK 要將 DOS 修改一下，但你不希望更改磁碟上的 DOS，因爲更改後便不能和普通 DOS 一樣由磁碟取出來。

INIT 的第二個分別就是它能在任何 SLOT 內 IN—ITIALIZE 一隻 RAM DISK。例如，你要使R—AM DISK對SLOT 4 有反應，可以打 INIT EMU，S4〔RETURN〕。然後又打 CATALOG，S4〔RE—TURN〕，你會發覺 WELCOME 程式消失了。

RAM DISK 可以插在任何沒有磁碟機的 SLOT 上，有時，就算已有週邊設置亦無妨。因爲 RAM DISK只會影响該個 SLOT 的磁碟機輸入和輸出，它亦可以插在印字機所用的 SLOT 裡，雖然印字機不受影响，但其他設備可能有影响。

INIT EMU 亦可以改變 RAM DISK的 VOLU—ME。RAM DISK 起初的 VOLUME NUMBER 是 212，如要改爲 100，要打入 INIT EMU，V100〔RETURN〕，當你貯起藏在模擬器的資料時，VOLUME NUMBER更加重要。

正如我們剛才提及，當使用 RAM DISK時 EMU是一個保留名字，在 INIT 指令，當 RAM DISK程式遇到 EMU作爲文件名稱時，便明白到這不是普通的 DOS 指令。除 INIT 指令外，其他 SAVE，LOAD，LOCK 和 UNLOCK指令也有同樣的情形。

EMU 之後跟著一個數字（EMU 1 至 EMU 8）亦是保留名字，如無法避免，可以用類似的名稱，例如 E—MULATOR，或 EMU TOO，則不是保留字。

SAVE EMU 和LOAD EMU 指令

這兩個指令是用來將 RAM DISK 的資料貯存入磁碟，或將 RAM卡的資料由磁碟取回。我們稱一隻貯有 RAM CARD 資料的磁碟做 EMU DISK，使它和傳統的磁碟不同。

SAVE 和 LOAD指令可以如常操作，除非跟著後面有以下九個保留名字其中一個，即 EMU，EMU 1，E—MU 2，EMU 3，EMU 4，EMU 5，EMU 6，EMU 7 或 EMU 8。如果你用其他文件名稱（例如：S—AVE ASTROSMASH），這指令可正常操作，寄存器內的程式便以這名稱存起。

打入以下指令，SAVE ASTROSMASH，S5〔RETURN〕，便將 ASTROSMASH 存入 RAM DI—SK。如打入 SAVE ASTROSMASH，S6，D1 就會將ASTROSMASH 貯入 SLOT 1，DRIVE 1（磁碟機 1）。

但如果你用任何一個保留字稱（例如，SAVE EMU 3，S6，D1〔RETURN〕），則16K的 RAM卡，（包括 RAM DISK 儲存的所有東西和 RAM DISK 的索引。）都當作一個單位，全部存入 SLOT 6，DRIVE 1 的磁碟內，而指令 LOAD EMU 3，S6，D1 的工作剛好相反，由 SLOT 6，DRIVE 1的磁碟內取資料存入 RAM 卡。

用 RAM卡進行 LOAD EMU或 SAVE EMU，你必須準備一隻特別泡製的磁碟，方法是運行 MAKE EM—UDISK程式。這程式將磁碟編排成 8 個儲存區（空間），每個面積都可以儲存 RAM卡的資料，這 8 個區，名爲E—MU 1，EMU 2，EMU 3 …… EMU 8。如你打入S—AVE EMU 3，S6，D1 指令，則 RAM卡的所有資料全部都儲入 SLOT 6，DRIVE 1 磁碟的第三個區。指令 LOAD EMU 3，S6，D1 的工作剛好相反，將該區的資料全部塡入 RAM 內。

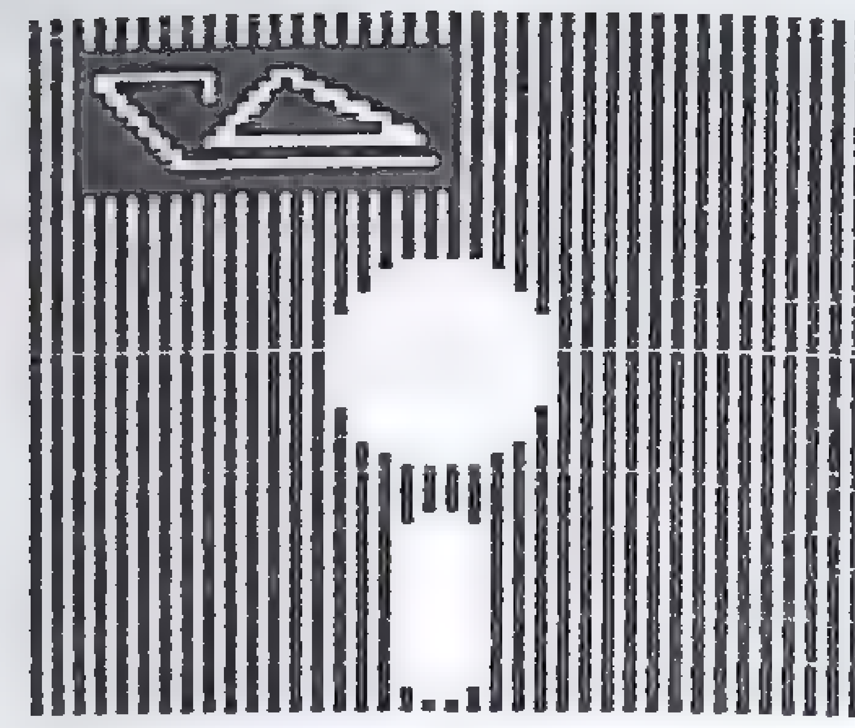
LOCK EMU 和UNLOCK EMU 指令

如果你偶一不愼，將 EMU貯入一隻普通磁碟上，該碟的目錄便會被破壞，使所有程式不能再用。爲避免這種錯誤，我們可以修改 RAM DISK，使它在儲入磁碟前，先做兩次測試。首先，它在磁碟上找尋 EMU，（在 VTOC 時 4至 6 BYTES）。

其次，它要檢查磁碟的 VOLUME NUMBER。如果它找不到 EMU，或 VOLUME NUMBER與現時的V—OLUME NUMBER不配合，它會發出「必」一聲，並停止 LOAD 或 SAVE。

你可以省略第一個測試，因爲這是 UNLOCK EMU 指令的功用。當你打入 UNLOCK EMU〔RETURN〕之後，RAM DISK 會將 RAM卡資料 SAVE 入 V—OLUME NUMBER 正確的磁碟上，亦可以由磁碟 LO—AD 出來。如果打 LOCK EMU〔RETURN〕，RAM DISK 會拒絕 SAVE 入未「準備好」的磁碟。那

將RAM卡作為磁碟機使用



麼，甚麼叫「準備好」的磁碟呢？它就是運行了 MAKE EMUDISK（列表了）的磁碟，其作用就是 INITIALIZE 。它的作用是先 UNLOCK EMU ，然後 SAVE EMU入磁碟的8個區。最後 LOCK EMU。

特別指令摘要

INIT EMU，Ss，Vv：清除一張16K卡，使它模擬在 SLOT S 的磁碟機， VOLUME NUMBER為V。

LOAD EMUa ，Ss ，Dd：由 SLOT S，DRIVE d 的磁碟內a區部份，LOAD RAM卡的資料。RAM DISK 的 VOLUME NUMBER 是取自磁碟的VOLUME NUMBER。LOAD EMU 即是 LOAP EMU 4。

SAVE EMUa ，Ss ，Dd：將 RAM卡的資料SAVE 入 SLOT S，DRIVE d的磁碟內區a部份。磁碟的 VOLUME NUMBER必定要與 RAM DISK的VOLUME NUMBER相配合。SAVE EMU 即是SAVE EMU 4。

UNLOCK EMU ：容許一隻未預先「準備好」的磁碟能進行 SAVE EMU和 LOAD EMU。

LOCK EMU ：拒絕使一隻未「準備好」的磁碟進行 SAVE EMU 和 LOAD EMU。

使用FID

APPLE 文件轉移程式 FID 不能直接將文件轉移到 RAM DISK 。然而，它可轉移到已定格式的 EMUDISK 的第4區。然後用 LOAD EMU〔RETURN〕指令整個轉移到 RAM DISK 。亦可以轉移到其他區域。例如區1，用 LOAD EMU〔RETURN〕和SAVE EMU 1〔RETURN〕。

脫離RAM DISK

脫離 RAM DISK （在列表4）是一個簡單程式，用以離開 RAM DISK，恢復原本的 DOS 。要把這程式貯入磁碟，打 BSAVE DISCONNECT RAM DISK，A$ 280，L$ 46 。它可以以鍵盤調用，或作為程式內其中一句語句，例如 10 PRINT CHR$（4）"BRUN DISCONNECT RAM DISK"。兩者都不會干擾任何 BASIC 程式或其變數，亦不中斷運行中的程式。脫離 RAM DISK 後，RAM DISK 原本的程式便會消失，所以如要保留，你必定要先 SAVE EMU。

錯誤訊息

當 RAM DISK 遇有錯誤（ ERROR ）時 ，便發出「必必」BEEPS 聲。由於空位有限，所以不一定打出錯誤的訊號。但是，錯誤通常都是很明顯易見的，以下是常見的錯誤情形：

1、當 SAVE EMU或 LOAD EMU到／由 EMU DISK 本身的 SLOT 時，會出現 I／O ERROR 訊息。補救方法是標明有磁碟機的 SLOT 。

2、在未有用 MAKE EMUDISK定格式的磁碟上， SAVE EMU或 LOAD EMU 出現 ERROR 。別忘記在新磁碟上使用 MAKE EMUDISK 程式作為起首第一步，亦可用 UNLOCK EMU 指令。（在使用 UNLOCK EMU 時要小心，EMU DISK 會強行寫在磁碟上，而不理會是否已有文件。）

3、當 SAVE EMU入碟時，如果它的 VOLUME NUMBER 和 RAM DISK 的當時 VOLUME 不相對應，便出現 VOLUME MISMATCH 訊息。應付方法是換另一隻磁碟，或使用 INIT EMU指令改變 RAM DISK 的 VOLUME 。

FILE緩衝器

RAM DISK 程式佔用3個普通 DOS文件緩衝器中的第三個，始於 $9AA6。它重新設定 DOS 以產生兩個文件緩衝器。這優點就是它不改變供用家使用的寄存器數量。通常，兩個緩衝已足夠，但偶然遇有 NO BUFFERS AVAILABLE 訊息出現，你可以在 BRUN RAMDISK 之後使用 MAXFILES 3 指令作為補救。

操作

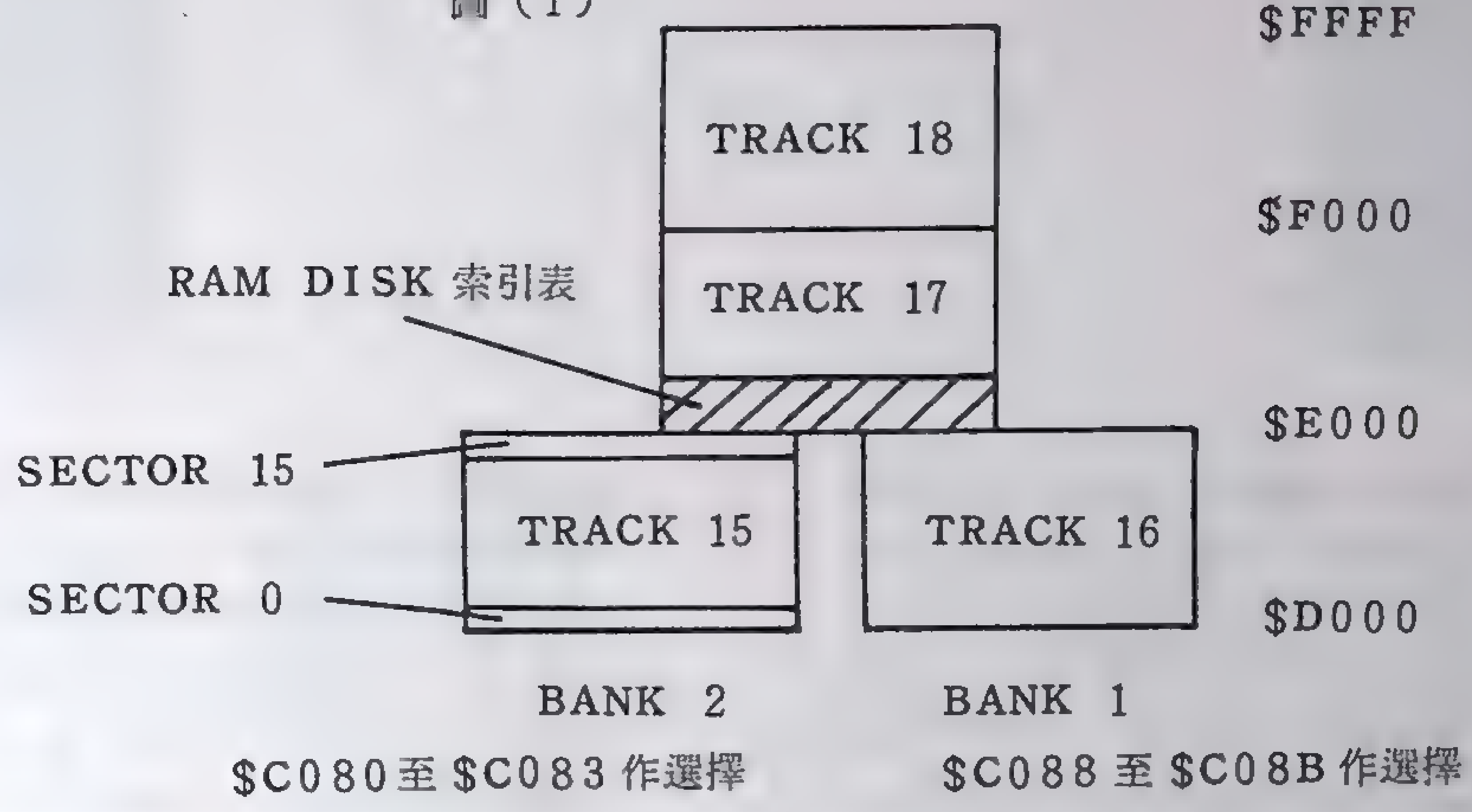
RAM DISK 的原理是將寄存器當作磁碟用。它用16K寄存卡上地址介乎＄D000至＄FFFF 的存貯器。這正是 BASIC 編譯程式和監督子程式 ROM。當16卡的存貯器「開著」時，BASIC和監督子程式「關掉」。當使用這些存貯器時，程式要負責「開著」存貯器，使用後，再「關掉」，才回復 BASIC 編譯程式的功用。

圖一是16K RAM卡上存貯器示意圖，與及一些將咭「開著」和「關掉」的地址。請注意寫資料在卡上的功能和使用存貯器的功能是分開的。例如，使用地址 $C081是用來寫在卡上，而 BASIC 仍然未被「關掉」。

為簡化起見，我們只用 RAM DISK上其中兩個地址，其中一個叫 LCONWRT，位於 $C083 ，它能「開着」卡並推行寫的功能。另一個是 LCOFFPRT，位於 $C082 ，它能「關掉」卡並保護它未被寫上。還有一個保險設施 ：你在寫上卡前，一定要在同一行用兩次 WRITE—

將RAM卡作爲磁碟機使用

圖（1）



位置C080，$C080，$C083，$C088 ，$C08B 控制記憶卡的開啓位置 C081， $C082，$C083，$C08A 控制記憶卡 的關閉。

雙數位置 WRITE PROTECT 記憶卡

單數位置可寫入記憶卡（假如有兩次）。

ENABLE 地址。

在圖一，還有一個特點，有兩個 BANK 的寄存器是賦予介乎 $D000—$DFFF 的寄存器。如果你用地址 LCONWRT 去「開著」卡，那一定是用其中一個 BANK 。如果用另一個 BANK 你要打入 LCONWRT＋8 。去「開著」這張卡。在列表內，你會發覺 RAM DISK 常用指數指示，例如 LDA LCONWRT，X去「開著」卡。如果X的值是零，即是選取一個 BANK ，如果X的值是 8 ，即是選另一個 BANK。

最後，圖一亦顯示16 K卡上的寄存器和磁碟上磁軌的相對值。在 $D000 至 $DFFF 兩個 BANK 相當於 TR—ACK 15和16，而 $E000至$FFFF相當於 TRACK 17和18。通常， TRACK 17 足夠填滿 CATALOG，但因爲16K卡比一張磁碟細小，所以 RAM DISK 需要的目錄位置細得多，我們預留了3個 SECTORS—— 足夠 21 個 FILENAME ， CATALOG 佔的位置是 $E100至 $E3FF ，相當於 TRACK 17， SECTORS 1至3 。

RAM DISK 的裝置

RAM DISK是 LOAD 進位於$ 9000的主要寄存器，改變少許 DOS 。這些改變，對於 RAM DISK 來說，是必須的，現在來研究一下。

RWTS（READ—WRITE—TRACK—SECTOR）是 DOS 的一部份，用來讀或寫 256 BYTES 到一個指定的 SECTOR 或由指定的 SECTOR讀／寫。所有D—OS 功能，無論如何複雜，都用 RWTS 來讀或寫磁碟，所以只要我們更改 RWTS 的起點，便能使它不做 RWTS 的正常工作，而送往 RAM DISK ，就可以改變所有 DOS 指令了。

RAM DISK 可以指示電腦檢查 DOS 指令，如果該指示是調用 RAM DISK，則 RAM DISK如常工作，否則，就轉往 RWTS 。

RAM DISK 有兩個方法知道 DOS 要調用它，如果 DOS 功能指明了插有 RAM卡的 SLOT，這個 SLOT 編號會出現在 $B7E9 的 INPUT—OUTPUT BLO—CK（IOB）內。 RAM DIRK會將這個數目和藏在 SLOT 的數目和藏在 SLOT 的數目相比較。請注意這個數目並非 SLOT 編號，而是 SLOT 編號的 16倍。

第二，如果 DOS 功能標明了其中一個保留名字（E—MU或 EMU 1至 EMU 8），則這個名會出現在始於 $AA75 的 FILE 名字緩衝器內， RAM DISK會在緩衝器內找尋 EMU，如果找到，就會執行該指令。

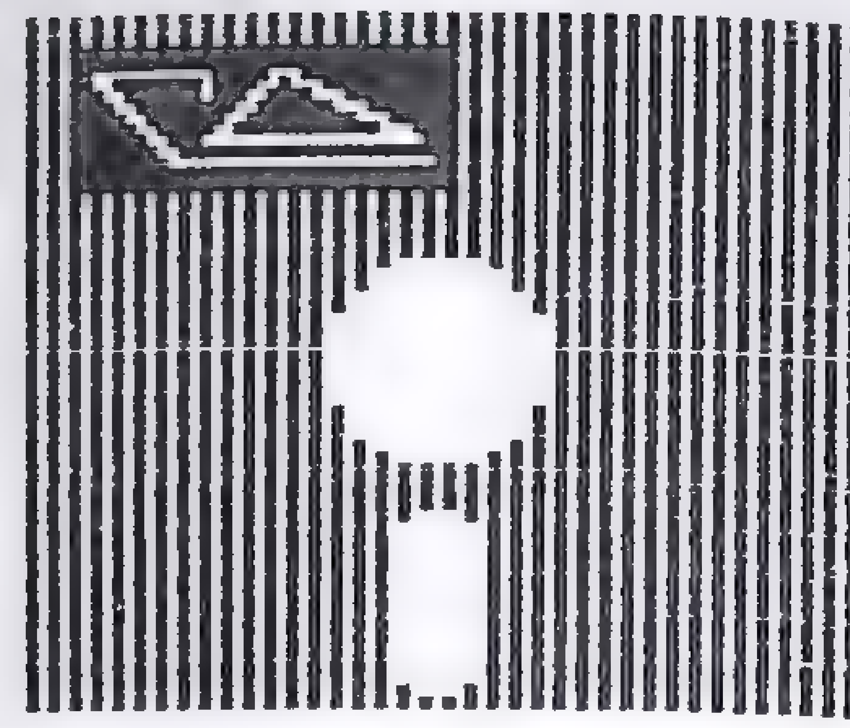
RAM DISK 重要功能

RAM DISK 亦有一些必備的功能，它一定要能 I—NITIAIZE RAM 卡（16 K 卡），把它清除，並在$E00 和 $E3FF 之間的建立 CATALOG 磁軌。它亦要處理 LOCK 和 UNLOCK指令，並在磁碟上的 VTOC 找尋 RAM DISK的名字（正如前面所述，這點工作雖然細微，但很重要，可以防止不慎將 RAM卡的資料存入已有文件的磁碟內）。但 RAM DISK最重要的功能是：（1）模擬轉送資料到（或自）磁碟的DOS 指令。（2）處理特別的 SAVE EMU和LOAD EMU 指令，以使傳送R—AM卡的資料。

模擬DOS 指令

模擬 DOS 指令是較容易的工作。我們要詳細討論，因爲我們要示範 RAM DISK 如何解決兩個難題——（1）它如何開啓存貯卡內正確的 BANK ，（2）它如何計算在 DOS 指令內 TRACK 和 SECTOR 的相應寄存位置。

將RAM卡作爲磁碟機使用



模擬 DOS 指令的代碼在 CHKSLT 開始（列表的行號 204）。RAM DISK讀出在 IOB 的 SLOT ，如果不是 RAM DISK的 SLOT 編號，便回轉 RWTS。然後， RAM DISK 由讀出 IOB 指令編號，指令O是找尋磁軌指令，它不須需調用 RAM DISK ，因此回轉返 RWTS 。指令4是 INITIALIZE。於是 INITIALIZE RAM DISK。指令1和2分別是讀和寫，於是存貯在磁碟上以備後用。

RAM DISK 讀出要用的磁軌。磁軌的範圍應介乎15至18（首尾包括在內），否則， RAM DISK會加4或減4，直至符合範圍。並將結果貯入 TRKBYTE，然後減15，使指數介乎0至3，放入X——寄存器，等到五個指令後，由 TBL 1 放一個數目入Y——寄存器。這個數目將會是0或8（視乎所需的磁軌），作用是指示 LDA LCONWRT，Y 選擇16K卡內正碼的 BANK 。

同時，RAM DISK 已把緩衝器指示字由 IOB 傳送到零頁上的一個位置，現在 RAM DISK要計算相對於 TRACK 和 SECTOR的存貯卡位置。每個相對 TRACK 的位置是在 TBL 2（用圖1的 TRACK 位置比較 TBL 2 內的4個數值）。

RAM DISK 的SECTOR

16個 SECTOR是由小至大存起來，例如， TRACK 17，SECTOR 0 由 $E000 開始，TRACK 17，SECTOR 由 $E0開始， TRACK 17，SECTOR 2 由 $E200開始。RAM DISK由 TBL 2 讀出 TRACK 位置，計算出 TRACK 和 SECTOR的開始位置。又經 OR 邏輯運算，得出高 BYTE 和低 BYTE （即零）結果，儲入零頁的指示字。現在所有工作準備好。 WRITE 和 READ 使用剛才零頁的指示字，傳輸 256 BYTES。READ 由16K卡傳資料到主寄存器，而 WRITE 則相反。RAM DISK 關掉16K卡，將本身 SLOT 編號和 VOLUME NUMBER 傳送到 IOB ，清除 CARRY FLAG ，以表示傳輸完畢。然後它用 RTS 指示，反回名爲 RWTS 的子程式。

SAVING 和LOADING

RAM DISK 進行特別的 SAVE 和 LOAD 功能始於列表的行號 287，這與模擬 DOS 指令有所不同，在模擬 DOS 指令，RAM DISK 完全脫離 RWTS ，永不回頭，但 SAVE 和 LOAD 則不同， RAM DISK 一定要由磁碟傳送資料，或傳送回磁碟，爲此，它要調用 RWTS （在TRKLP，行號 421）。事實上，它調用了 RWTS 64次！即每傳轉一個 SECTOR 調用一次。

任何調用 RWTS 的子程式一定要建立一個 IOB 以控制 RWTS。至於 SAVE 和 LOAD 功能， RAM DISK使用自己的 IOB 而不是 DOS 的 IOB（這 IOB 在列表尾，行號 538），因此 RAM DISK 使用 SAVE 和 LOAD 時就比較快些。

RAM DISK 可以將8張 RAM卡寄存量貯寄在一隻磁碟上。每張 RAM卡資料佔4個 TRACKS。請參閱以下表一。它選擇的 TRACK 就視乎指令後的 EMU的數目（EMU 1 至 EMU 8）TRACK 數目的計算在 MAKETRK，LINE 338。

程式(一)

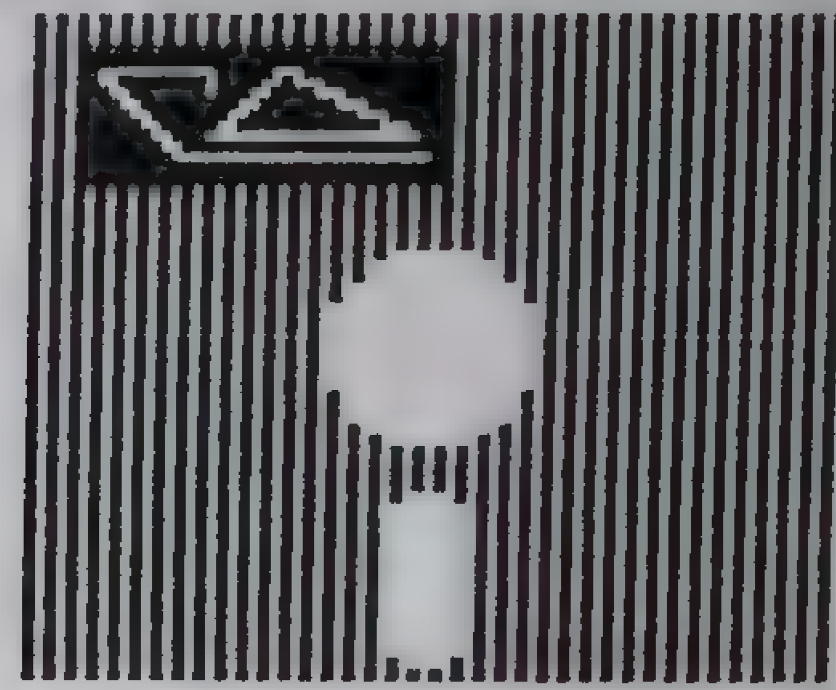
```
*9000L

9000-    A0 02       LDY   #$02
9002-    B9 55 90    LDA   $9055,Y
9005-    99 B7 B7    STA   $B7B7,Y
9008-    88          DEY
9009-    10 F7       BPL   $9002
900B-    A0 02       LDY   #$02
900D-    B9 58 90    LDA   $9058,Y
9010-    99 2A BD    STA   $BD2A,Y
9013-    88          DEY
9014-    10 F7       BPL   $900D
9016-    A9 60       LDA   #$60
9018-    8D 6B A5    STA   $A56B
901B-    A9 0E       LDA   #$0E
```

表(一)

Name	Tracks	Catalog track
DOS	0-2	
EMU1	3-6	5
EMU2	7-10	9
EMU3	11-14	13
EMU4 or EMU	15-18	17
EMU5	19-22	21
EMU6	23-26	25
EMU7	27-30	29
EMU8	31-34	33

將RAM卡作爲磁碟機使用



```
901D-   8D 93 B2    STA   $B293
9020-   A9 5B       LDA   #$5B
9022-   85 3C       STA   $3C
9024-   A9 90       LDA   #$90
9026-   85 3D       STA   $3D
9028-   A9 A9       LDA   #$A9
902A-   85 3E       STA   $3E
*L

902C-   A9 92       LDA   #$92
902E-   85 3F       STA   $3F
9030-   A9 A6       LDA   #$A6
9032-   85 42       STA   $42
9034-   A9 9A       LDA   #$9A
9036-   85 43       STA   $43
9038-   A0 00       LDY   #$00
903A-   20 2C FE    JSR   $FE2C
903D-   A9 02       LDA   #$02
903F-   8D 57 AA    STA   $AA57
9042-   A9 80       LDA   #$80
9044-   8D 00 9D    STA   $9D00
9047-   A9 9A       LDA   #$9A
9049-   8D 01 9D    STA   $9D01
904C-   20 D4 A7    JSR   $A7D4
904F-   20 53 9C    JSR   $9C53
9052-   4C 7E 9B    JMP   $9B7E
9055-   20 A7 9A    JSR   $9AA7
9058-   20 AF 9C    JSR   $9CAF
905B-   FF          ???
*9AA6L

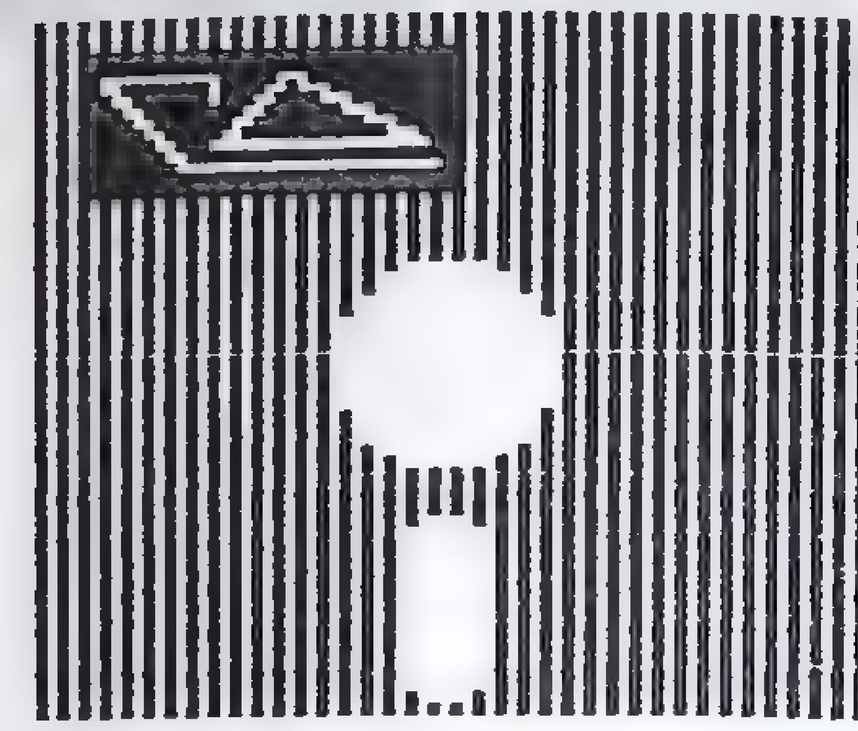
9AA6-   FF          ???
9AA7-   84 48       STY   $48
9AA9-   85 49       STA   $49
9AAB-   A0 1D       LDY   #$1D
9AAD-   A2 19       LDX   #$19
9AAF-   A9 A0       LDA   #$A0
9AB1-   D9 75 AA    CMP   $AA75,Y
9AB4-   D0 2C       BNE   $9AE2
9AB6-   88          DEY
9AB7-   CA          DEX
9AB8-   10 F7       BPL   $9AB1
9ABA-   A0 02       LDY   #$02
9ABC-   B9 B8 9C    LDA   $9CB8,Y
9ABF-   D9 75 AA    CMP   $AA75,Y
9AC2-   D0 1E       BNE   $9AE2
9AC4-   88          DEY
9AC5-   10 F5       BPL   $9ABC
9AC7-   AC 5F AA    LDY   $AA5F
9ACA-   C0 0C       CPY   #$0C
9ACC-   F0 0B       BEQ   $9AD9
*L

9ACE-   C0 0E       CPY   #$0E
9AD0-   F0 07       BEQ   $9AD9
9AD2-   C0 05       CPY   #$05
9AD4-   B0 0C       BCS   $9AE2
9AD6-   4C 5B 9B    JMP   $9B5B
9AD9-   98          TYA
9ADA-   E9 0E       SBC   #$0E
9ADC-   8D A6 9A    STA   $9AA6
9ADF-   4C 7E 9B    JMP   $9B7E
9AE2-   AD E9 B7    LDA   $B7E9
9AE5-   CD BB 9C    CMP   $9CBB
9AE8-   D0 6E       BNE   $9B58
9AEA-   AD F4 B7    LDA   $B7F4
9AED-   F0 58       BEQ   $9B47
9AEF-   C9 04       CMP   #$04
9AF1-   D0 03       BNE   $9AF6
9AF3-   4C 7B 9B    JMP   $9B7B
9AF6-   48          PHA
9AF7-   AD EC B7    LDA   $B7EC
9AFA-   C9 13       CMP   #$13
*L

9AFC-   90 04       BCC   $9B02
9AFE-   E9 04       SBC   #$04
9B00-   B0 F8       BCS   $9AFA
9B02-   C9 0F       CMP   #$0F
9B04-   B0 04       BCS   $9B0A
9B06-   69 04       ADC   #$04
9B08-   90 F8       BCC   $9B02
9B0A-   8D 78 04    STA   $0478
9B0D-   E9 0F       SBC   #$0F
9B0F-   AA          TAX
9B10-   AD F0 B7    LDA   $B7F0
9B13-   85 3E       STA   $3E
9B15-   AD F1 B7    LDA   $B7F1
9B18-   85 3F       STA   $3F
9B1A-   BC BC 9C    LDY   $9CBC,X
9B1D-   B9 83 C0    LDA   $C083,Y
9B20-   B9 83 C0    LDA   $C083,Y
9B23-   BD C0 9C    LDA   $9CC0,X
9B26-   0D ED B7    ORA   $B7ED
9B29-   85 47       STA   $47
*L

9B2B-   A9 00       LDA   #$00
9B2D-   85 46       STA   $46
9B2F-   68          PLA
9B30-   4A          LSR
9B31-   B0 0B       BCS   $9B3E
9B33-   A0 00       LDY   #$00
9B35-   B1 3E       LDA   ($3E),Y
9B37-   91 46       STA   ($46),Y
9B39-   88          DEY
9B3A-   D0 F9       BNE   $9B35
9B3C-   F0 09       BEQ   $9B47
9B3E-   A0 00       LDY   #$00
9B40-   B1 46       LDA   ($46),Y
9B42-   91 3E       STA   ($3E),Y
9B44-   88          DEY
9B45-   D0 F9       BNE   $9B40
9B47-   2C 82 C0    BIT   $C082
```





```
9B4A-    AD BB 9C    LDA    $9CBB
9B4D-    8D F7 B7    STA    $B7F7
9B50-    AD E0 9C    LDA    $9CE0
*L

9B53-    8D F6 B7    STA    $B7F6
9B56-    18          CLC
9B57-    60          RTS
9B58-    4C 04 BD    JMP    $BD04
9B5B-    98          TYA
9B5C-    D0 2A       BNE    $9B88
9B5E-    A9 10       LDA    #$10
9B60-    2C 65 AA    BIT    $AA65
9B63-    D0 04       BNE    $9B69
9B65-    A9 50       LDA    #$50
9B67-    D0 03       BNE    $9B6C
9B69-    AD E9 B7    LDA    $B7E9
9B6C-    8D BB 9C    STA    $9CBB
9B6F-    AD EB B7    LDA    $B7EB
9B72-    C9 FE       CMP    #$FE
9B74-    90 02       BCC    $9B78
9B76-    A9 00       LDA    #$00
9B78-    8D E0 9C    STA    $9CE0
9B7B-    20 53 9C    JSR    $9C53
9B7E-    A9 00       LDA    #$00
*L

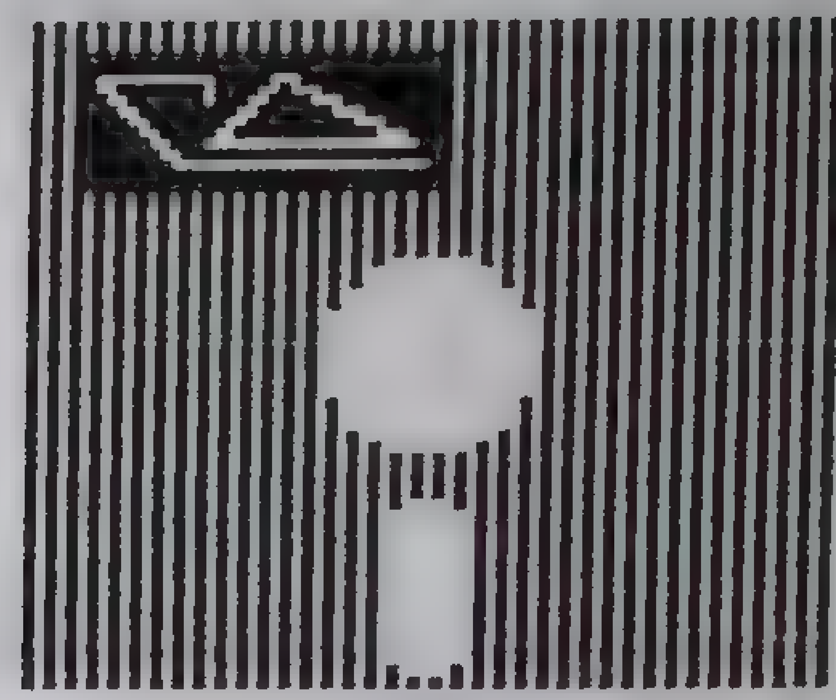
9B80-    85 48       STA    $48
9B82-    20 EA A2    JSR    $A2EA
9B85-    4C 8B 9F    JMP    $9F8B
9B88-    4A          LSR
9B89-    8D ED 9C    STA    $9CED
9B8C-    AD 78 AA    LDA    $AA78
9B8F-    C9 B1       CMP    #$B1
9B91-    90 08       BCC    $9B9B
9B93-    C9 B9       CMP    #$B9
9B95-    B0 04       BCS    $9B9B
9B97-    29 0F       AND    #$0F
9B99-    D0 02       BNE    $9B9D
9B9B-    A9 04       LDA    #$04
9B9D-    0A          ASL
9B9E-    0A          ASL
9B9F-    18          CLC
9BA0-    69 01       ADC    #$01
9BA2-    85 1F       STA    $1F
9BA4-    AD ED 9C    LDA    $9CED
9BA7-    C9 01       CMP    #$01
*L

9BA9-    D0 04       BNE    $9BAF
9BAB-    A9 00       LDA    #$00
9BAD-    F0 03       BEQ    $9BB2
9BAF-    AD E0 9C    LDA    $9CE0
9BB2-    8D E4 9C    STA    $9CE4
9BB5-    2C A6 9A    BIT    $9AA6
9BB8-    10 17       BPL    $9BD1
9BBA-    A5 1F       LDA    $1F
9BBC-    8D EC B7    STA    $B7EC
9BBF-    20 04 BD    JSR    $BD04
9BC2-    B0 76       BCS    $9C3A
9BC4-    A0 05       LDY    #$05
9BC6-    B9 BB B3    LDA    $B3BB,Y
9BC9-    D9 DA 9C    CMP    $9CDA,Y
9BCC-    D0 60       BNE    $9C2E
9BCE-    88          DEY
9BCF-    D0 F5       BNE    $9BC6
9BD1-    AD E9 B7    LDA    $B7E9
9BD4-    CD BB 9C    CMP    $9CBB
9BD7-    F0 55       BEQ    $9C2E
*L

9BD9-    8D E2 9C    STA    $9CE2
9BDC-    AD EA B7    LDA    $B7EA
9BDF-    8D E3 9C    STA    $9CE3
9BE2-    A4 1F       LDY    $1F
9BE4-    C8          INY
9BE5-    8C E5 9C    STY    $9CE5
9BE8-    A9 00       LDA    #$00
9BEA-    8D E9 9C    STA    $9CE9
9BED-    A9 E1       LDA    #$E1
9BEF-    85 48       STA    $48
9BF1-    A9 9C       LDA    #$9C
9BF3-    85 49       STA    $49
9BF5-    A9 03       LDA    #$03
9BF7-    85 1E       STA    $1E
9BF9-    A4 1E       LDY    $1E
9BFB-    BE BC 9C    LDX    $9CBC,Y
9BFE-    BD 83 C0    LDA    $C083,X
9C01-    BD 83 C0    LDA    $C083,X
9C04-    B9 C0 9C    LDA    $9CC0,Y
9C07-    09 0F       ORA    #$0F
*L

9C09-    8D EA 9C    STA    $9CEA
9C0C-    A9 0F       LDA    #$0F
9C0E-    8D E6 9C    STA    $9CE6
9C11-    20 04 BD    JSR    $BD04
9C14-    B0 24       BCS    $9C3A
9C16-    CE EA 9C    DEC    $9CEA
9C19-    CE E6 9C    DEC    $9CE6
9C1C-    10 F3       BPL    $9C11
9C1E-    CE E5 9C    DEC    $9CE5
9C21-    C6 1E       DEC    $1E
9C23-    10 D4       BPL    $9BF9
9C25-    AD EF 9C    LDA    $9CEF
9C28-    8D E0 9C    STA    $9CE0
9C2B-    4C 34 9C    JMP    $9C34
9C2E-    2C 82 C0    BIT    $C082
9C31-    20 3A FF    JSR    $FF3A
9C34-    2C 82 C0    BIT    $C082
9C37-    4C 7E 9B    JMP    $9B7E
9C3A-    2C 82 C0    BIT    $C082
9C3D-    A0 0D       LDY    #$0D
*L
```

將RAM卡作爲磁碟機使用



```
9C3F-    B1 48       LDA   ($48),Y
9C41-    4A          LSR
9C42-    4A          LSR
9C43-    4A          LSR
9C44-    4A          LSR
9C45-    A0 03       LDY   #$03
9C47-    88          DEY
9C48-    F0 03       BEQ   $9C4D
9C4A-    4A          LSR
9C4B-    90 FA       BCC   $9C47
9C4D-    B9 F2 9C    LDA   $9CF2,Y
9C50-    4C 85 B3    JMP   $B385
9C53-    A9 FF       LDA   #$FF
9C55-    85 47       STA   $47
9C57-    A0 00       LDY   #$00
9C59-    84 46       STY   $46
9C5B-    98          TYA
9C5C-    2C 83 C0    BIT   $C083
9C5F-    2C 83 C0    BIT   $C083
9C62-    91 46       STA   ($46),Y
*L

9C64-    2C 8B C0    BIT   $C08B
9C67-    91 46       STA   ($46),Y
9C69-    88          DEY
9C6A-    D0 F3       BNE   $9C5F
9C6C-    C6 47       DEC   $47
9C6E-    A5 47       LDA   $47
9C70-    C9 D0       CMP   #$D0
9C72-    B0 E7       BCS   $9C5B
9C74-    A0 0D       LDY   #$0D
9C76-    B9 C4 9C    LDA   $9CC4,Y
9C79-    99 74 E0    STA   $E074,Y
9C7C-    88          DEY
9C7D-    10 F7       BPL   $9C76
9C7F-    A0 07       LDY   #$07
9C81-    B9 D2 9C    LDA   $9CD2,Y
9C84-    99 30 E0    STA   $E030,Y
9C87-    88          DEY
9C88-    10 F7       BPL   $9C81
9C8A-    A9 7A       LDA   #$7A
9C8C-    8D 27 E0    STA   $E027
*L

9C8F-    A0 06       LDY   #$06
9C91-    B9 DA 9C    LDA   $9CDA,Y
9C94-    99 00 E0    STA   $E000,Y
9C97-    88          DEY
9C98-    10 F7       BPL   $9C91
9C9A-    A9 11       LDA   #$11
9C9C-    A0 02       LDY   #$02
9C9E-    8D 01 E3    STA   $E301
9CA1-    8C 02 E3    STY   $E302
9CA4-    88          DEY
9CA5-    8D 01 E2    STA   $E201
9CA8-    8C 02 E2    STY   $E202
9CAB-    2C 82 C0    BIT   $C082
9CAE-    60          RTS
9CAF-    EC BB 9C    CPX   $9CBB
9CB2-    F0 03       BEQ   $9CB7
9CB4-    DD 8C C0    CMP   $C08C,X
9CB7-    60          RTS
9CB8-    C5 CD       CMP   $CD
9CBA-    D5 50       CMP   $50,X
*L

9CBC-    00          BRK
9CBD-    08          PHP
9CBE-    08          PHP
9CBF-    08          PHP
9CC0-    D0 D0       BNE   $9C92
9CC2-    E0 F0       CPX   #$F0
9CC4-    FF          ???
9CC5-    FF          ???
9CC6-    00          BRK
9CC7-    00          BRK
9CC8-    FF          ???
9CC9-    FF          ???
9CCA-    00          BRK
9CCB-    00          BRK
9CCC-    FF          ???
9CCD-    F0 00       BEQ   $9CCF
9CCF-    00          BRK
9CD0-    FF          ???
9CD1-    FF          ???
9CD2-    11 01       ORA   ($01),Y
*L

9CD4-    00          BRK
9CD5-    00          BRK
9CD6-    23          ???
9CD7-    10 00       BPL   $9CD9
9CD9-    01 04       ORA   ($04,X)
9CDB-    11 03       ORA   ($03),Y
9CDD-    C5 CD       CMP   $CD
9CDF-    D5 D4       CMP   $D4,X
9CE1-    01 60       ORA   ($60,X)
9CE3-    01 00       ORA   ($00,X)
9CE5-    11 00       ORA   ($00),Y
9CE7-    FB          ???
9CE8-    B7          ???
9CE9-    00          BRK
9CEA-    00          BRK
9CEB-    00          BRK
9CEC-    00          BRK
9CED-    00          BRK
9CEE-    00          BRK
9CEF-    00          BRK
*L

9CF0-    60          RTS
9CF1-    01 08       ORA   ($08,X)
9CF3-    07          ???
9CF4-    04          ???
```

程式(二)

```
RAM DISK

TEXTFILE DEMO
310  IF ( PEEK (47032) <  > 167) THEN
      PRINT  CHR$ (4)"BRUN RAM DI
     SK"
320 D$ =  CHR$ (4): DIM DR$(1):DR
     $(1) = "> EMU <":DR$(0) = ">
      DISK  <"
330  FOR DR = 0 TO 1:SL = 6 - DR:
      HOME : VTAB 2: PRINT "** CO
     PYRIGHT 1984 BY COMPUTING AG
     E  **": VTAB 8: PRINT "BUILD
     ING A TEXTFILE ON "DR$(DR): GOSUB
     380: NEXT DR
340  HOME : VTAB 4: PRINT "TYPE A
      NUMBER FROM 1 TO 26": PRINT
     : INPUT "0 WILL END THE PROG
     RAM ";RC: ON RC = 0 GOTO 370
     : ON RC < 1 OR RC > 26 GOTO
     340
350  FOR DR = 0 TO 1: PRINT : PRINT
      CHR$ (7)"RETRIEVING  RECORD
      "RC" FROM "DR$(DR): GOSUB 3
     90: PRINT  CHR$ (7)RC$: PRINT
     "ANY KEY TO CONTINUE  ";: GET
     A$: PRINT : PRINT : NEXT DR
360  PRINT "SEE THE DIFFERENCE?":
      FOR I = 1 TO 1000: NEXT I: GOTO
     340
370  HOME : PRINT "GOODBYE": PRINT
     D$"DELETETEXT,S6": END
380  PRINT D$"OPENTEXT,L20,S"SL: FOR
     I = 1 TO 26: PRINT D$"WRITET
     EXT,R"I:C = I + 64: PRINT  CHR$
     (C) CHR$ (C)"  "DR$(DR)"  " CHR$
     (C) CHR$ (C): NEXT I: PRINT
     D$"CLOSE": RETURN
390 SL = 6 - DR: PRINT D$"OPENTEX
     T,L20,S"SL: PRINT D$" READTE
     XT,R"RC: INPUT RC$: PRINT D$
     "CLOSE": RETURN
```

程式(三)

```
MAKE EMUDISK

310  IF ( PEEK (47032) <  > 167) THEN
      PRINT  CHR$ (4)"BRUN RAM DI
     SK"
320  HOME : VTAB 6: PRINT "THIS P
     ROGRAM WILL PREPARE A DISK F
     OR USE": PRINT "     WITH RA
     M DISK (EMU)"
330  PRINT : FLASH : PRINT " WARN
     ING: ";: NORMAL : PRINT " TH
     IS PROGRAM WILL DESTROY ALL"
     : PRINT "     INFORMATION ON
      THE DISK!"
340  VTAB 22: PRINT "** COPYRIGHT
      1984 BY COMPUTING AGE **": VTAB
     14: HTAB 8: INPUT "SLOT?
     ";SL$: ON SL$ = "" GOTO 400:
     SL =  VAL (SL$): IF SL < 1 OR
     SL > 6 THEN 340
350  VTAB 16: HTAB 8: INPUT "DRIV
     E?   ";DR$: ON DR$ = "" GOTO
     400:DR =  VAL (DR$): IF DR <
     1 OR DR > 2 THEN 350
360  VTAB 18: HTAB 8: INPUT "VOLU
     ME?  ";VO$: ON VO$ = "" GOTO
     400:VO =  VAL (VO$): IF VO <
     1 OR VO > 254 THEN 360
370  PRINT : PRINT "  INSERT THE
     DISK AND HIT <RETURN>  ";: GET
     A$: PRINT : IF A$ <  >  CHR$
     (13) THEN 400
380 D$ =  CHR$ (4): PRINT D$"INIT
     EMU,V"VO: PRINT D$"INITHELLO
     ,S"SL",D"DR",V"VO: PRINT D$"
     UNLOCKEMU"
390  FOR EM = 1 TO 8: PRINT D$"SA
     VEEMU"EM",S"SL",D"DR: NEXT E
     M: PRINT D$"LOCKEMU": HOME :
      VTAB 8: PRINT "PREPARE ANOT
     HER DISK ?   ";: GET A$: PRINT
     A$: IF A$ =  CHR$ (89) THEN
     320
400  HOME : PRINT "GOODBYE": END
```

程式(四)

```
*280L

0280-   A0 02       LDY   #$02
0282-   B9 C0 02    LDA   $02C0,Y
0285-   99 B7 B7    STA   $B7B7,Y
0288-   88          DEY
0289-   10 F7       BPL   $0282
028B-   A0 02       LDY   #$02
028D-   B9 C3 02    LDA   $02C3,Y
0290-   99 2A BD    STA   $BD2A,Y
0293-   88          DEY
0294-   10 F7       BPL   $028D
0296-   A9 4C       LDA   #$4C
0298-   8D 6B A5    STA   $A56B
029B-   A9 11       LDA   #$11
029D-   8D 93 B2    STA   $B293
02A0-   AD 82 C0    LDA   $C082
02A3-   A9 A0       LDA   #$A0
02A5-   8D 75 AA    STA   $AA75
02A8-   20 EA A2    JSR   $A2EA
02AB-   A9 03       LDA   #$03
02AD-   8D B1 AA    STA   $AAB1
*L

02B0-   A9 D3       LDA   #$D3
02B2-   8D 00 9D    STA   $9D00
02B5-   A9 9C       LDA   #$9C
02B7-   8D 01 9D    STA   $9D01
02BA-   20 D4 A7    JSR   $A7D4
02BD-   4C 8B 9F    JMP   $9F8B
02C0-   20 00 BD    JSR   $BD00
02C3-   DD 8C C0    CMP   $C08C,X
```

太陽系模擬系統

探測六大行星移動位置

對微型電腦來說，應用它的高解像度螢幕是無止境的，其中用來模擬一些圍繞我們的東西是最好不過的了，本文是討論一下我們的太陽系首六顆星球。

太陽系中的每一顆行球都以隋圓軌跡來運轉，距離太陽愈近，便需時愈少以完成軌跡。以水星（MERCURY）爲例，它是最接近太陽的球體，完成一次軌跡需時八十八天。而土星則需時二十九年，是六顆行星之中最遠的。正因爲它們運行是以隋圓軌跡的，因此與太陽的距離是不斷的改變中，利用 JOHANN KEPLER（1571—1630）的第二定律，我們可以計算出圖一的隋圓軌跡中由W點至R點的時間。如你所見，RV此綫段是連結太陽與星球的，而此長度會經常變更，開始時 RV= SW，而 SWR這個面積可以用公式表

1.) Area = $\frac{ab}{2}$ (H-e sin H).

而 a是 MAJOR AXIS，b是 MINOR AXIS， E是 ECCENTRICITY（偏心距）$=\frac{c}{a}$，H（見圖2）是中心點至q點，q點是在以a爲半徑的圓上一點。

利用公式①可計算出H角，我們將①式重組成爲

2). H-e sin H = $\frac{area * 2}{ab}$

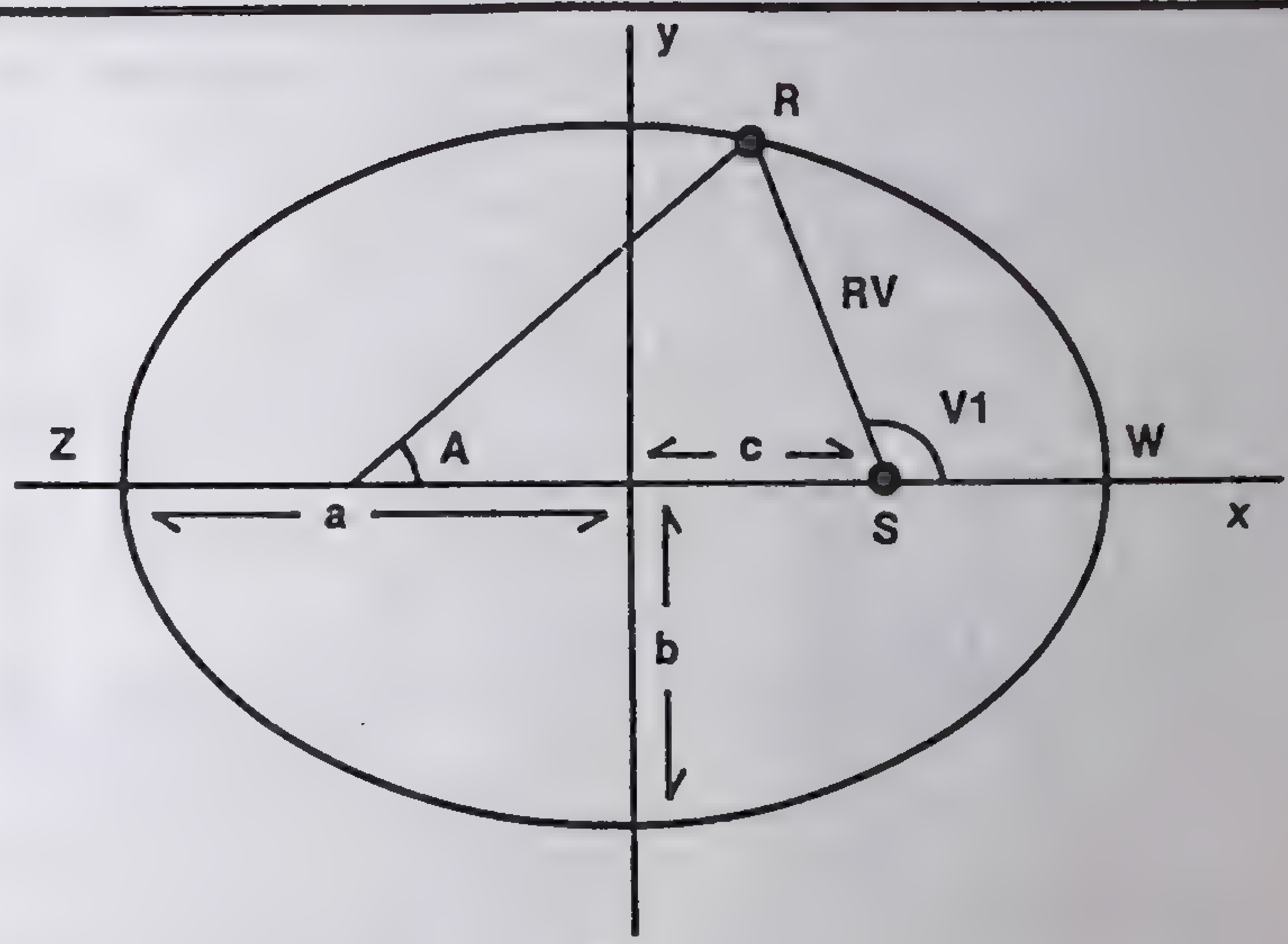
我們不能將②式簡化，因爲H和 SIN—H 是不可分解因式的，因此我爲着方便起見便將式除以 360 度來求出圖①中A角，以公式③

3.) RV = 2a - (P/1 + e(cos(180-A)))

我們可以用它來得出太陽及星球間的A角，若用另一公式：

4.) cos V1 = $\frac{R - RV}{RV*e}$

在上式中的P是由焦點至隋圓的垂直綫 $=a(1-e^2)$將A角遞增，我們便



圖(1)

	Sidereal revolution in days	Distance from Sun in million miles max.	min.	Longitude of perhelion in degrees	Eccentricity
Mercury	87.969	43.403	28.597	77.1	.2056
Venus	224.701	67.726	66.813	131.3	.0068
Earth	365.256	94.555	91.445	102.6	.0167
Mars	686.980	154.936	128.471	335.7	.0934
Jupiter	4332.125	507.046	460.595	13.6	.0478
Saturn	10825.863	937.541	838.425	95.5	.0555
Uranus	30676.15	1859.748	1699.331	172.9	.0503
Neptune	59911.13	2821.686	2760.386	58.5	.0066
Pluto	90824.2	4551.386	2756.427	223.0	.2548

表(一)

可以求得繪在螢幕上。

不過採用角A亦會出現問題，就是增加角A數値同時， SWR 面積（圖1）不增加，因此會導出錯誤。因此將①式與③式連結得出：

5). cos H = $\frac{a-RV}{ae}$

以求出用角A所產生的誤差。

現在我將本程式介紹一下，因爲要縮短程式，我只包括了六個行星在內，但若閣下要增加行星數目時，可參考 LISTING 1加入行星繪出的次序是由太陽始，水星、金星、地球、火星、木星、土星，你可選擇其中的任何一星球來顯示，直至同時六個。

你亦可繪出任何一日的星球位置，或從一個特定日子以時間長度計算，例如一九八四年一月一日後 100 天的位

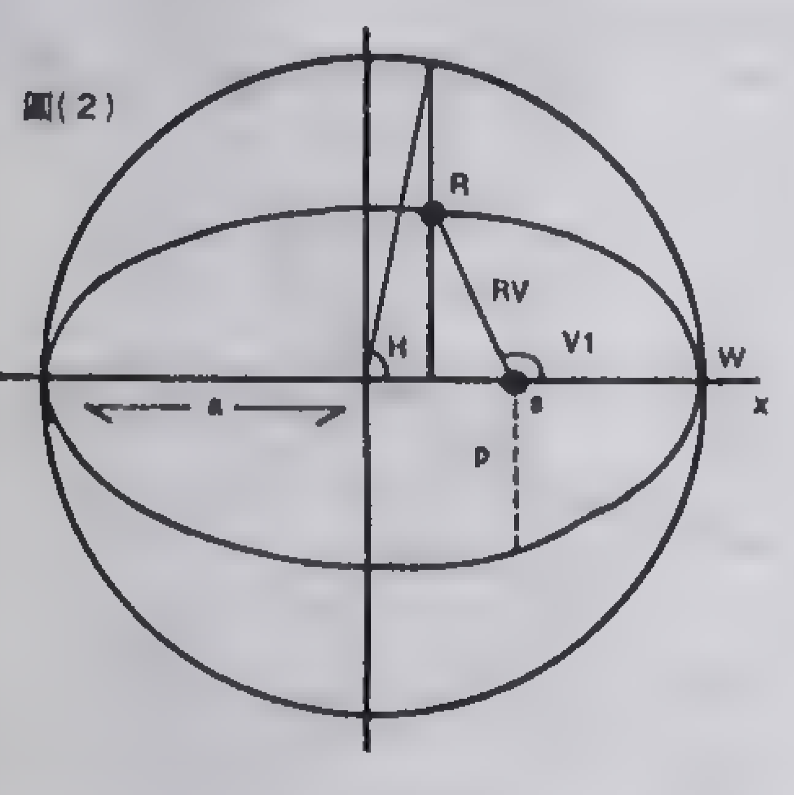
探測六大恒星移動位置

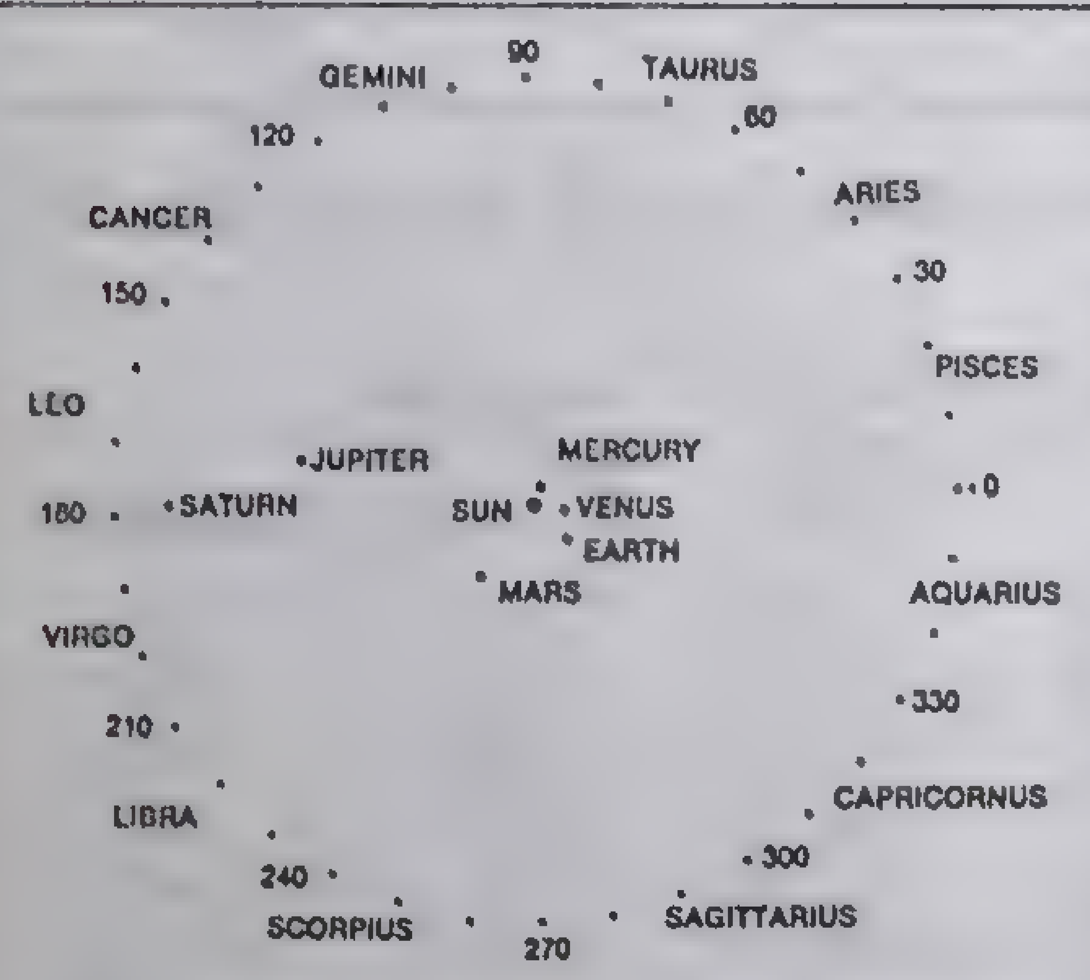
置。亦可以繪出以例如總時間長度爲 900 天，每一點相距爲 30 天，亦可以用單點或連綫形成顯示。

圖 3 是一個本程式的例子。其他的星座名，行星名字是用作參考的。

圖 4 是繪出水星、金星、地球的例子，本例是從五月二十九日（ 150 日）後的 44 天內以每點相隔 4 天連續繪出。而本程式的精確度亦十分令人滿意。

圖(2)





水星、金星及地球在由一九八〇年五月廿九日（第 150 日）開始的四十四日。每隔四日繪一點。

圖(3)



圖(4)

在一九八〇年八月十一日（第224日）的六大行星位置圖。

```
]PR#0

]LIST

10  REM  ************************
     ***
12  REM  *
     *
14  REM  * SOLAR SYSTEM SIMULATIO
     N *
18  REM  *
     *
20  REM  *         SOLAR
     *
22  REM  *
     *
28  REM  *
     *
29  REM  ************************
     ***
30  REM
80 EL = 1461.024: GOTO 1000
90  REM  (100-110) PLOT X AND Y V
     ALUES
100  HPLOT X,Y
110  RETURN
150  REM  (200-300) CALCULATE THE
      X AND YPLANET POSITIONS
200 D = Z -  INT (Z / SRD) * SRD
205  REM  D IS FOR DAYS
210 B = Q - (D / SRD * Q2)
220 RV = A - (P / (1 + E *  COS (
     B)))
225  REM  RV IS THE RADIUS VECTOR
      OR DISTANCE FROM THE SUN TO
      THE PLANET
230 V = PE / RV - EZ
240  IF V =  > 1 THEN V = VL
245  IF V =  <  - 1 THEN V =  - V
     L
250 V1 =  -  ATN (V /  SQR ( - V *
     V + 1)) + T
255  REM  V1 IS THE ANGLE THAT TH
     E PLANET LIES FROM THE SUN.
     THE 0 POINT BEING AT THE RIG
     HT, INCREASING COUNTERCLOCKW
     ISE.
260  IF D < SRD / 2 THEN V1 = Q2 -
     V1
270 V1 = V1 + J
280 X =  COS (V1) * RV:Y =  -  SIN
     (V1) * RV * FA
290 X = X * TT + X1:Y = Y * TT +
     Y1
300  RETURN
900  REM  (1000) DISPLAY PRIMARY
     PAGE, SET TEXT MODE
1000  POKE  - 16300,0: POKE  - 16
     303,0
1010 T = 1.5708
1020 Q = 3.14159265
1030 Q2 = 6.2831853
1040 VL = .99999999
1050 FA = 29 / 32
1055  REM  FA IS THE RATIO OF X T
     O Y TO PLOT A CIRCLE AN THE
     APPLE INSTEAD OF AN OVAL
1060 X1 = 140:Y1 = 96
1700  HOME : PRINT : PRINT : PRINT
     : PRINT
1800  PRINT "DO YOU WANT TO DISPL
     AY "
1810  PRINT : PRINT "THE SAME PLA
     NETS AS YOUR LAST RUN"
1815  PRINT : INPUT "Y OR N ";A$
1820  PRINT : PRINT
1830  IF A$ = "N" THEN 2000
1840  IF A$ <  > "Y" THEN 1800
1850  IF S1 <  > 0 THEN 4000
1855  PRINT : PRINT
1860  PRINT : PRINT "YOU HAVEN'T
     PICKED THE PLANETS YET"
1870  PRINT : PRINT : PRINT
2000  PRINT "CHOOSE THE PLANETS Y
     OU WANT TO DISPLAY"
2005  PRINT
2010  PRINT "ENTER A 1 FOR YES AN
     D A 0 FOR NO"
2011  PRINT
2012  REM (2020-2079) GET SPECIFI
     C VALUES FOR EACH PLANET
2013  REM  S1=ORBITAL PERIOD: P1=
     A1*(1-E1*E1)/2
2014  REM  E1=ECCENTRICITY: U1=P1
     /E1:K1=1/E1
2015  REM  REM A1= MINIMUM AND MA
     XIMUM DISTANCE FROM THE SUN
2016  REM  J1=LONGITUDE OF PERIHE
     LION IN RADIANS
2017  REM  W1= DAYS FROM 0 DEGREE
     S TO PERIHELION FOR 1980
2018  REM  TT=SCALING FACTOR TO U
     SE FULL PLOTTING AREA IF SEL
     ECTED PLANETS ARE DISPLAYED
2020  INPUT "DISPLAY MERCURY   ";
     ME
2021 S1 = 87.969
2022 E1 = .2056
2023 A1 = 43.403 + 28.597
2024 P1 = A1 * (1 - E1 * E1) / 2
2025 K1 = 1 / E1
2026 U1 = P1 / E1
2027 J1 = 77.1 * Q / 180
2028 W1 = 37.58 + EL
2029  IF ME = 1 THEN TT = 2.3
2030  INPUT "DISPLAY VENUS     ";
     VE
2031 S2 = 224.701
2032 E2 = .0068
2033 A2 = 67.726 + 66.813
2034 P2 = A2 * (1 - E2 * E2) / 2
2035 K2 = 1 / E2
2036 U2 = P2 / E2
2037 J2 = 131.3 * Q / 180
2038 W2 = 140.5 + EL
2039  IF VE = 1 THEN TT = 1.5
2040  INPUT "DISPLAY EARTH     ";
     EA
2041 S3 = 365.256
2042 E3 = .0167
2043 A3 = 94.555 + 91.445
2044 P3 = A3 * (1 - E3 * E3) / 2
2045 K3 = 1 / E3
2046 U3 = P3 / E3
2047 J3 = 102.6 * Q / 180
2048 W3 =  - 3 + EL
2049  IF EA = 1 THEN TT = 1.05
2050  INPUT "DISPLAY MARS      ";
     MA
2051 S4 = 686.980
2052 E4 = .0934
2053 A4 = 154.936 + 128.471
2054 P4 = A4 * (1 - E4 * E4) / 2
2055 K4 = 1 / E4
2056 U4 = P4 / E4
2057 J4 = 335.7 * Q / 180
2058 W4 = 289 + EL
2059  IF MA = 1 THEN TT = .6
2060  INPUT "DISPLAY JUPITER   ";
     JU
2061 S5 = 4332.125
2062 E5 = .0478
2063 A5 = 507.046 + 460.595
2064 P5 = A5 * (1 - E5 * E5) / 2
2065 K5 = 1 / E5
2066 U5 = P5 / E5
2067 J5 = 13.6 * Q / 180
2068 W5 = 1604 + EL
```

探測六大恒星移動位置

```
2069  IF JU = 1 THEN TT = .19
2070  INPUT "DISPLAY SATURN    ";
     SA
2071 S6 = 10825.863
2072 E6 = .0555
2073 A6 = 937.541 + 838.425
2074 P6 = A6 * (1 - E6 * E6) / 2
2075 K6 = 1 / E6
2076 U6 = P6 / E6
2077 J6 = 95.5 * Q / 180
2078 W6 = 2115 + EL
2079  IF SA = 1 THEN TT = .1
3900  HOME : PRINT : PRINT
4000  PRINT : PRINT "DO YOU WANT
     ": PRINT
4010  INPUT "POINT (0) OR CONTINU
     OUS (1) PLOTS ";TY
4015  IF TY <  > 0 AND TY <  > 1 THEN
     4000
4020  PRINT : PRINT : PRINT
4030  PRINT : PRINT "DO YOU WANT
     TO START AT": PRINT
4040  PRINT "A SPECIFIC DATE (0)
     ": PRINT
4050  INPUT "OR THE BEGINNING OF
     THE YEAR (1) ";DT
4051  IF DT <  > 0 AND DT <  > 1 THEN
     4020
4052  IF DT = 1 THEN 4060
4053  PRINT : PRINT : PRINT
4054  INPUT "ENTER # OF DAYS SINC
     E JAN 0,1984 ";DE
4057 Z1 = DE
4060  PRINT : PRINT : INPUT "ENTE
     R # OF DAYS TO PLOT ";DN
4070  PRINT : PRINT : PRINT
4080  INPUT "ENTER # OF DAYS BETW
     EEN PLOTS ";DA
4082  IF DA <  > 0 THEN 4800
4084  PRINT : PRINT
4086  PRINT "0 NOT ALLOWED": GOTO
     4070
4090  REM  4800 INIT HIGH RES, FU
     LL SCREEN, PAGE 2
4800  HGR2
4802  REM  (4805-4860) PLOT REFER
     ENCE POINTS AND OUTER 10 DEG
     REE CIRCLE
4805  HCOLOR= 3
4810 X = 140:Y = 96: GOSUB 100
4811 X = 141:Y = 96: GOSUB 100
4815 X = 248:Y = 96: GOSUB 100
4820  FOR L1 = 0 TO Q2 STEP 1 / 3
     6 * Q2
4830 X = X1 +  COS (L1) * 105.9
4840 Y = Y1 -  SIN (L1) * 105.9 *
     FA
4850  GOSUB 100
4860  NEXT L1
4900  REM  (5100-5140) SET UP VAL
     UES FOR MERCURY AND PLOT
5100  IF ME = 0 THEN 5200
5110 A = A1:P = P1:E = E1:PE = U1
     :EZ = K1:SRD = S1:J = J1:W =
     W1:Z = Z1 + W
5120  GOSUB 200:F1 = X:G1 = Y
5125  IF TY = 1 THEN 5140
5130 X = M1:Y = N1: HCOLOR= 0: GOSUB
     100
5140 X = F1:Y = G1:M1 = X:N1 = Y:
     HCOLOR= 3: GOSUB 100
5190  REM  (5200-5240) SET UP VAL
     UE FOR VENUS AND PLOT
5200  IF VE = 0 THEN 5300
5210 A = A2:P = P2:E = E2:PE = U2
     :EZ = K2:SRD = S2:J = J2:W =
     W2:Z = Z1 + W
5220  GOSUB 200:F2 = X:G2 = Y
5225  IF TY = 1 THEN 5240
5230 X = M2:Y = N2: HCOLOR= 0: GOSUB
     100
5240 X = F2:Y = G2:M2 = X:N2 = Y:
     HCOLOR= 3: GOSUB 100
5290  REM  (5300-5240) SET UP VAL
     UES FOR EARTH AND PLOT
5300  IF EA = 0 THEN 5400
5310 A = A3:P = P3:E = E3:PE = U3
     :EZ = K3:SRD = S3:J = J3:W =
     W3:Z = Z1 + W
5320  GOSUB 200:F3 = X:G3 = Y
5325  IF TY = 1 THEN 5340
5330 X = M3:Y = N3: HCOLOR= 0: GOSUB
     100
5340 X = F3:Y = G3:M3 = X:N3 = Y:
     HCOLOR= 3: GOSUB 100
5390  REM  (5400-5440) SET UP VAL
     UES FOR MARS AND PLOT
5400  IF MA = 0 THEN 5500
5410 A = A4:P = P4::E = E4:PE = U
     4:EZ = K4:SRD = S4:J = J4:W =
     W4:Z = Z1 + W
5420  GOSUB 200:F4 = X:G4 = Y
5425  IF TY = 1 THEN 5440
5430 X = M4:Y = N4: HCOLOR= 0: GOSUB
     100
5440 X = F4:Y = G4:M4 = X:N4 = Y:
     HCOLOR= 3: GOSUB 100
5490  REM   (5500-5540) SET UP VA
     LUES FOR JUPITER AND PLOT
5500  IF JU = 0 THEN 5600
5510 A = A5:P = P5:E = E5:PE = U5
     :EZ = K5:SRD = S5:J = J5:W =
     W5:Z = Z1 + W
5520  GOSUB 200:F5 = X:G5 = Y
5525  IF TY = 1 THEN 5540
5530 X = M5:Y = N5: HCOLOR= 0: GOSUB
     100
5540 X = F5:Y = G5:M5 = X:N5 = Y:
     HCOLOR= 3: GOSUB 100
5590  REM  (5600-5640) SET UP VAL
     UES FOR SATURN
5600  IF SA = 0 THEN 6000
5610 A = A6:P = P6:E = E6:PE = U6
     :EZ = K6:SRD = S6:J = J6:W =
     W6:Z = Z1 + W
5620  GOSUB 200:F6 = X:G6 = Y
5625  IF TY = 1 THEN 5640
5630 X = M6:Y = N6: HCOLOR= 0: GOSUB
     100
5640 X = F6:Y = G6:M6 = X:N6 = Y:
     HCOLOR= 3: GOSUB 100
6000 Z1 = Z1 + DA
6100  IF Z1 > DE + DN THEN 7000
6200  GOTO 5100
7000 X = 279:Y = 190: GOSUB 100: INPUT
     A$
7050  REM  (7000) PLOT POINT 297
     190 TO INDICATE END OF SIMUL
     ATION THEN WAIT FOR INPUT OF
      ANY CHARACTER TO START AGAI
     N
7100 Z1 = 0:DE = 0
7200  GOTO 1000
```

APⅡ六代同堂
微電腦之進化史

陸生

自從AP Ⅱ微電腦在本港推出，流行以來，屈指一數，到現在大約已有兩年多的時間了，在這兩年後的今天，筆者大胆在這裡作一個小小的總結。在這段時間中，話短不短，說長不長，其AP Ⅱ所經歷的驚濤駭浪，可以編寫一部歷史，自AP Ⅱ在本港面世後，它最初所帶來的，是一份驚喜若狂的感覺。及至中段的時期，它所帶來的，是澎湃洶湧的熱潮。以至現在的AP Ⅱ，它由昔的光輝而歸於平淡了。在這三個時期中，AP Ⅱ的機種（GENERATION）也就順應潮流所需而作出了改變。以下的文章，筆者嘗試去根據不同時期的AP Ⅱ，而去描劃出每一個時期的變更及其所帶來的影响。

(1) 第一代——套件形式出現

第一代的AP Ⅱ微電腦，大約於兩年前在本港面世。當時本港的業餘電子愛好者還在熱衷於擴音機及均衡器的製作。第一部AP Ⅱ還只是以套件的形式在深水埗的鴨寮街出售，其時AP Ⅱ的套件包括有主機板，電源供應器，鍵盤等，而外殼只是一些質料低劣的塑膠啤製而成，既不美觀又不能受太大的壓力，而全套套件的售價却要三千多元。這個時間，可以說是AP Ⅱ的雛型時期，故亦可以說這是第一代。這時期的AP Ⅱ，零件及綫路板大多來自台灣，只有少量是日本及本港的製品。它們的質數極其參差，一個好的AP Ⅱ，它的綫路板對於正常的操作佔了極重要的地位，其次它的電源供應器也是影响操作的一個環節，零件供應的不足也是一個頭痛的問題。在整體結構上，它的容量爲48 K BYTE。擴展插座也只是一至三個，電源供應器之＋5 V電壓也只可以供一至兩張介面卡使用，遑論磁碟機或印字機的介面配合使用了。這時期的製成品，鍵盤只是一個狀似普通打字機的打字盤，既沒有指示燈，又沒有分離的數字鍵，彈性不足及容易出錯，要寫一個程式，單是鍵盤所引致的錯字，就可以改上半天，眞是惱人。

周邊裝置的配合，只是限於螢幕監視器或經PAL卡配合的電視機作爲影像的輸出。而其他的介面卡，如RAM卡，Z－80卡，80－COL卡等等還未普遍，就算有，價錢之高令人咋舌。其輸入／輸出的裝置，只是一部普通的錄音機，全部程式沓用錄音帶作爲儲存體，錄音機的層次式檔案傳輸，可靠性低及效率慢，雖是惱人但却是無可奈何。軟件的供應，絕大部份爲電腦遊戲，只有少數的爲實用的軟件。這時間的用家也只是極少數，因爲要配合基本的組合，其一般的價錢非要五千元不可呢！

(2) 第二代——成品出售

第一代大約持續了四個月左右，第二代的APⅡ微電腦便推出了。隨著了用家的需要及綫路上的修正，硬件及軟件的配合要來得完整及齊備，技術上的錯誤也就跟著修正改良而減少。在這時期的AP Ⅱ，產品也大多來自台灣，而再不是以套件的形式出現，而是整套製成品出售。其整體的結構有了一定的改良，如48 K BYTE的記憶容量，8個完整的擴展插座，全部零件配以IC插座，令到維修方面更容易，喇叭的音量放大了，令到玩電腦遊戲時更有實感。電源供應器也可以供應2.5A的電流，換句話說，它可以供應全機的擴展介面卡，包括了磁碟機、印字機等等。鍵盤的設計也有了些突破，就是在〝A－SDF〞及〝JKL；〞等鍵上有了凹位，令到打字時可以這樣爲依歸（H－OME KEY），另外又在鍵盤的左下方另加裝了指示燈，令到用家知道他的AP Ⅱ正在操作，除了這些改良外，它的外殼也有了突破，它再不是用塑膠啤出來的，而是用合成纖維倒膜而成，機身質地較結實，可以承受螢幕監視器或電視機的重量，甚至一至二個磁碟機的重量也是沒有問題，如著名的〝海利〞殼便是，又因爲玩AP Ⅱ微電腦的用家已較爲普遍，AP Ⅱ的價錢也就直綫下降，這時售賣AP Ⅱ微電腦的商店也不再限於鴨寮街，AP Ⅱ微電腦專賣店就如雨後春筍般開設，而深水埗的高登商場也就搖身一變成爲AP Ⅱ微電腦的集散場及總滙處，其熱鬧的情況眞是一時無兩。

隨著用家的要求及普遍，不同的介面卡也就陸續推出了，如語言卡，RA－M卡，INTEGER BASIC卡，ROM卡，80－COL卡，Z－80卡，磁碟機介面卡，PAL卡等等，因著了這些介面卡，AP Ⅱ的用家所面對的AP世界又更廣濶了。在周邊裝置方面，程式的儲存也不局限於錄音帶了，廉價的磁碟機面世後，幾乎取代了錄音機的地位，這時磁碟機及螢幕監視器，可以說是必需品，軟件的性質也主要是遊戲方面，而應用軟件也正在陸續推出，一時間抄錄之風大盛，無形中助長了磁碟片的銷售量，價錢也因而大跌。舉個例子說，初時DYSAN的磁碟片一盒十張爲280元，而跌至170元左右，其110元之差距皆由惡性商業競爭所致，而得益者自然是用家了。

(3) 第三代——鍵盤大大改良

第二代大約持續了半年左右，便被更新及加以收良的第三代取代了位置。這時間的AP Ⅱ，在外形上並沒有太大的改變，但最明顯的分別就是在於鍵盤上，在鍵盤的左下方的指示燈改爲透明的十字燈，零售商也常常標榜著這個改變，作爲宣傳及推廣第三代的AP Ⅱ微電腦的口號。除了初期的十字燈外，至中期時，更把整個鍵盤略作縮小而左移，而騰空了的右邊位置，則加了15個數字的數字鍵盤（NUMERIC KEY PAD）這個改變可以大大地提高了一般會計工作的數據輸入效率，但對一般用家而言却似乎沒有多大的用處，甚至有些用家認爲有多餘的感覺呢！但這是見仁見智的說法而己！正當零售商在大力宣傳這些新設備之際，鍵盤的設計又再有了新突破，這個末期的突破就是鍵

AP II 微電腦之進化史

盤內藏有功能鍵（ FUNCTION K—EYS ）。它的原理就是在十幾個鍵子上先行定了一些 APPLESOFT 的常用命令，如〝 SAVE〞，〝LOAD〞，〝CATALOG〞等等，把這些資料儲存在一隻 EPROM 中，當按著 C—TRL 鍵及任何一個指定的鍵子時，那特定的命令便會執行，確是令到編寫程式時方便得多。

這時期的 AP II，主機板來自日本，而主機板又可分爲手銲板及錫爐板，手銲板由人手逐粒零件銲接在主機板上，質數便有所參差，如假銲點，松香過多而形成接觸不良者比比皆是，主機板一旦有問題，便令到維修工作益見困難，售價自是比較廉宜。錫爐板則是工廠式的製作，把所有零件全插上主機板上，再整塊板被推進錫爐中噴錫，錫點自然粒粒均勻，絕無松香過多或假銲點的情形出現，對 AP II 的操作更臻完美，售價比較昂貴。至於電源變壓器方面又有了進一步的改良，除了加大了的電流量（可達 7 AMP）外，還加了零負荷時的切斷電源的保護，對 AP II 主機板又多了一個保障，這時著名的如日本的多孔黑牛（HK＄1,000 ），美國銀牛（HK＄550）便是。這時主機板也曾流行過用速度高（ 80 NS ） 的金 RAM，但因價格高昇而令用家却步。

介面卡及軟件的供應也因用家的大量需要而益見繁多及普及。這時的介面卡，如 REAL TIME CLOCK，RS—232 C SERIAL INTERF—ACE，COMMUNICATION 卡，SOUND BOARD，80K RAM EXPANSION（FOR VISICA—LC）等等也推出了，不同階層的用家又去利用這些介面卡去開發他們的五彩繽紛，永無休止的世界， AP II 已開始進入狂熱的階段。但這個熱潮並未有維持很久便因它的商業版權法而接連遭受到海關的檢控，這時著名的 AP II 微電腦集散地，如深水埗的高登及黃金商場、灣仔的海運商場等的零售商便紛紛收到了票控，搜查及有關的硬件，軟件及書籍被沒收的情形，經此一役後，各大商場又不復當日的熱鬧了，那種門庭冷落，景象蕭條的情景與昔日的光輝成了一強烈的對比！

(4) 第四代——綫路設計錯

因爲版權法及惡性的商業競爭，頭腦靈活的商人便把第三代再加以改良而令到新一代的 AP II 有了一個新的面貌。在鍵盤的外型方面，它與前幾代的沒有甚麼明顯的分別，但是在功能方面，它却加多了很多功能鍵，在指示燈的左下方近指示燈的位置，它新增了一個〝FUNC〞鍵，利用這個鍵與其他的按鍵相配合便可產生不同的命令，在 AP—PLESOFT 中，它可以有 94 個不同的配合，從而產生 94 個不同的命令，如〝VTAB〞，〝HTAB〞，〝XD—RAW〞等等。在 CP/M 方面，它同樣可以產生 94 個不同的命令，如〝BEEP〞，〝SYSTEM〞等等。爲了方便用家易於使用，製造商把這些命令令印在兩張紅、藍的對照表上，給用家方便查閱，除了鍵盤的改進外，主機板上也有了不小的改變，最明顯的就是 RAM 及 ROM 方面，這時的讀寫記憶體已擴展到 64 K，而唯讀記憶體擴展至 24 K，在這 24 K 中，它包括了 INTERGER BASIC 的 IN—TERPETER 在內，只要在開機後打〝INT〞便可進入 INTERGER BASIC 中，方便無比。可能是綫路改變時有些錯誤，第四代推出後不久，便有用家發現它與 CP／M 的 INTE—RFACE 有問題，在執行應用軟件〝WORD STAR〞時不能正常操作，有操作不暢順的情形，究其原因，可能是主機板與 Z—80 卡在時間（ TIMI—NG）上不能配合而產生錯誤，這或是製造商在主機板綫路的變動上未有週詳的考慮而急於推出市面所致。第四代風行了不多時便消聲匿跡了。

在這個時期，印字機的裝設已不再爲具規模的公司所專有，隨着用家的普遍要求，印字機牌子之多，品種之衆，功能之相近，價錢之大衆化眞叫選購者舉棋不定。

(5) 第五代——多一枚Z —80

第四代煙沒了不多時，第五代又推出了。這次的改變，重點放在主機板上，最引人注目的，這是那 8 個擴展插座，分別用黑、啡、紅、橙、黃、綠、藍、紫 8 個顏色順次編排，它的好處就是令用家可以依擴展插座的顏色而辨別介面卡的編配法，不但省時，而且可以避免不必要的錯誤。其次改變的，就是在主機板上加多了一隻 Z—80 的微處理器，換句話來說，它省却了 Z —80 的介面卡，又因爲 Z—80 是與主機板直接連接，免却了因 TIMING 不相配合的問題。因爲加了 Z— 80的原故，它沒有足夠的位置去容納那 12 KROM，於是製造商便用 3 隻 2764 的 EP—ROM 取代了原本那 6 隻的 2716 EPROM，把這 2764 的 EPROM 介面卡永遠插在零號插座中，這便解決了地方不足的問題。除了主機板上的改變外，鍵盤還多了十個由用家設計的字鍵（ PF KEYS），這十個字鍵由一組獨立的電池維持，有效期可達五年之久，這十個鍵可以隨意更改或抹掉，其他的字盤同時又具有自動重覆的特性，當某一個字鍵被按下超過一秒鐘的 時候，它便會自動印出相關的英文字母，大小楷也可以，這對鍵盤的設計又是一個突破了。

(6) 第六代——附加多項特性

與第五代差不多同時推出的便是第六代，這一代的產品可以說是第三代的擴展，除了 64 K RAM 之外，它還附加了不少新的特性：

（1）內藏 MINI MONITOR，可直接編寫 ASSEMBLER 語言。

（下轉第42頁）

新介面科技的探討

本期有關蘋果及HP最新電腦就是一個開始

新一代私人電腦硬件及軟件正開始打進市場，這些系統操作更容易兼且比以前的類型更具生產力，就算不是電腦專家，在使用新私人電腦時仍感到日常工作非常舒適寫意，並會發覺甚少出錯。

"新介面科技"啓動了硬件軟件的新發展，基本上是減少了使用者需要牢記的事項，亦能非常有效地使用電腦系統。在硬件方面，指點裝置（POINTING DEVICES）假如「老鼠仔」，觸摸螢幕及高解像繪圖顯示，簡化了用家及電腦系統間的溝通。軟件則提供「完整視窗」，「多個視窗」而由指點裝置發出號令。這些特徵結合起來，可確保用家能集中注意力於人的工作上，而毋須顧慮電腦如何操作。

新科技的樣本，部份經已面世，包括軟件／硬件組合，如蘋果麗莎及HP的150，軟件如VISICORP'S VISION及MICROSOFT'S WINDOWS。

關於新介面科技的歷史，首先要致謝蘭克施樂PARC研究中心的努力。不過，介面科技的源頭要追溯至DOUGLAS ENGLEBART的研究，他用電腦來爭論人類的智慧。ENGLEBART首先發明類似現在的「老鼠仔」及「複合視窗」，加入文件編輯的設計上。

XEROX PARC研究工作始於一九七一年，進行DYNABOOK計劃，目標是要將如記事簿大小的私人電腦，令任何人，甚至是兒童，都可應用於日常工作上，而每個人都想擁有一部。XEROX ALTO 私人電腦是用作建立起樣版DYNABOOK系統。

雖然受硬件的限制，令DYNABOOK未能作商業性大量生產，但新介面科技很多特點都可以追源自DYNABOOK。DYNABOOK 計劃其中一個重要的產品，是SMALLTALK語言，這種概念引發了很多人擴大它，廣泛應用於電腦行業上，包括新的SMALLTALK版本，並發明了重叠視窗。

硬件的配合

這些系統很獨特，令軟件硬件的設計有很好的連繫。硬件很多特點是考慮到操作更複雜軟件的需要。系統採用16－BIT處理器（IMBPC INTEL 8088 或HP 150，或麗莎的MOTORROLA 68000）。整個軟件需要最少512 K BYTE RAM及二分一M BYTE 磁碟儲存（通常是一部WINCHESTER 磁碟）。

將這些機器稱爲微型電腦實有誤導性。系統裝備有記憶系統，一部硬碟，及16－BIT處理器，可以接近現代迷你電腦的處理速度。

硬件其餘的特點，高解像度BIT MAP 繪圖顯示器及指點裝置，增加I／O（輸出／輸入）濶度。高解像繪圖顯示器很重要，因要它促使電腦傳達資料予使用者更有效率，比低解像及單是文字的顯示器更勝一籌。諺語有云：「一張圖片抵上千言萬語」，我們實難估計顯示器的價值。新系統利用繪圖的優點，以加強與用家的溝通。它提供穩定的視覺回饋以幫助帶引用家及改進整個系統的質素。它可以將文件以不同大小的字體顯示出來。並可將文件按比例顯示出來，而印出來亦是一同大小，這種技巧被稱爲"你看到甚麽就可得到甚麽"。這減少使用者繁複工作，就可得到心目中所需要的文件。

指點裝置是輸入補充繪圖顯示器。沒有這種裝置，會造成拖延。指計裝置已廣泛地存在——例如老鼠仔，觸摸螢幕，軌球（TRACK BALLS），歡樂棒，光筆及繪圖板——「老鼠仔」是最值得推薦。它比其他設計生產成本較廉宜。「老鼠仔」是高解像設備，它可以點高解像螢幕上一個光點（PIXEL），就算你的手離開「老鼠仔」，它仍保持功能。當你需要經常轉動於鍵盤及指點裝置之間時，它這項特點就顯得妙用無窮。

軟件的配合

新介面科技最重要及最革命性的地方是軟件。它連用硬件可以製造很多類型的系統。從用家的觀點來看，軟件有三個明顯易見的組合：多個重叠視窗；利用指點裝置發出命令；應用程式時，可以簡單及持久地將資料聯繫其他程式。

視窗及指點設計的應用是整個新系統的中心，而很明顯比舊系統有很明顯的改良。當系統在操作時，螢幕就像工作枱，可以劃分爲幾個長方形地區，或稱之爲「視窗」。普通一個程式就利用一個視窗，透過視窗，程式就可以與使用者溝通。

LISA 系統，將目標物，例如檔案，用圖畫符號"ICON"代表。ICON 概念與道路交通標誌的相同——是希望用最短的時間傳送訊息。ICONS是令致高解像繪圖能力成爲重要東西的原因之一。如果螢幕不適宜顯示ICONS，可以將視窗改爲顯示文字。

在螢幕上時刻都有游標連繫指點裝置。例如，當使用者移動老鼠仔時，游標就會按比例地移動。游標代表了使用者注意力的集中點，它可以是一個箭咀以指示螢幕上準確的一點。

所有輸入，在箭咀的尖端附近進行，是游標的"熱點"，游標令使用者插入及選擇更形順暢。選擇的目標物，視乎MENU 所下的命令，可以是ICON，一段文字，或者是繪圖影像，都可以容易獲到。目標物便可以被抽離而操作。這過程須提供重要的視覺回饋。在舊式的傳統介面設計，缺點是經常會忽略了資料。沒有回饋，實無可能知道系統究竟是否對使用者的要求作出反應，所以使用者要重覆命令，以確保機器清楚明白指示。這並不理想，尤其是那些有繼續印字緩衝間的系統，更易起誤會。

老鼠仔可以很方便地同時處理兩項操作，例如它可以在

螢幕上抽取文字或圖案來達到理想的安排。一些系統更提供〝KNOBS〞及〝LEVERS〞以控制螢幕上的目標物。在LISA當游標在視窗邊緣拖行〝直降機〞時，視窗更可以捲動。

老鼠仔另一應用稱爲〝橡筋圖案〞（RUBBER－BAND GRAPHS）。例如是橡筋視窗，窗的一角固定位置而對角接上老鼠仔游標。視窗仍是長方型，但是四邊的四度則拉長了，變成窗的形狀伸展。這種視覺上的效果是很有趣。此技巧可以用於繪圖程式，綫、弧型等等可以像橡筋般拉長縮短。

這些系統的索引目錄主要是令使用者有更多的選擇，對工作有幫助，而極力避免了因有大量無關係的東西，而引致感到迷惑困擾。基於此原因，索引表通常儘量簡短而剔除了不適用的選擇。

新系統軟件最寶貴的能力是它可以在兩個應用程式中互相交流資料。令使用者縱觀程式，利用此能力去剪裁資料。（蘋果新出的MACINTOSH便有此功能，請參考本期有關蘋果新電腦文章）。

爲了使一個應用程式與另個互爲使用，須要採用有規格的方法傳送資料。一種是透過兩個視窗（由有應用程式）傳送資料。任何資料脫離視窗以後，是由工作枱經理程式（它的〝視窗〞是在窗之下的工作桌灰色地帶，負責監視兩個視窗間之資料傳送。這種能力是新系統設計時最艱巨的項目之一。LISA及VISION，都未能克服此困難，做到兩個程式互傳資料。不過，兩間公司已聲言他們的產品未來可以支持這種操作。

新軟件的心理學及哲學

發展新軟件，大多數的決定是建基於哲學原則，在設計〝SMALLTALK SYSTEM〞時，很多哲學原則已首先被分析。而稍後XEROX STAR計劃作提煉昇華。

用家介面的定義是經過長時期的研究始能訂定。定義決定了新系統的性格及習慣，故此，所有在系統運行的程式都應符合綱領。其主要需要克服的事項有四個：

第一，限制程式與使用者互爲影响途徑的數目。迫令所有程式服從這些原則，使用者在學習系統時，可以儘量少記需要注意事項。

第二，選擇最佳的互爲影响的形式，及堅持所有程式用它。這方面，很多哲學原則應用於此。

第三，程式需要可以繪圖，同時又可處理老鼠仔，是非常複雜。通常，需要寫上編釋語言去達到適當的表現。（LISA及VISI ON就是如此），但採用了綱領後，原來的軟件生產商可以提供這些設備。當考慮到LISA的工作桌經理——10MEGABYTE密碼資源——就知道對軟件發展件節省了幾多時間及金錢。

第四：所有互爲影响作用都是準確規格令應用程式互傳資料，更易於控制，這是設計新系統最困難的問題。

這些系統的哲學原則大同小異，並在持久性的概念上打轉，概念是電腦所運用的資料客體，被視爲我們所想像的眞實物質客體。當形容客體任何的運作時，主要是有兩個組合：主體（SUBJECT），它是客體本身（OBJECT），及動詞（VERB），任何動作加諸於客體上者，一如在眞實世界，同樣的動詞可以應用於很多不同主體，所以同樣的行動可以應用到系統內不同的資料客體。

移動客體或目標物到不同的視窗，無論是句子，繪圖影像，一組數字都是用同樣的途徑。

首先，目標物先被選擇，跟着動詞〝CUT〞送予目標物，使它離開原有的窗，使用者無須顧慮被移動的目標物是文字，繪圖或數字。當然在用家介面的程式，較之用不同命令施於目標物的程式，來得更爲複雜。

簡化與持久性是二爲一體，利用〝POINT AND C—LICK SYNTAX〞，使用者用游標指向目標物及〝C—LICKS〞（按下老鼠仔的鍵鈕或將手指離開螢幕）去選擇它，最複雜類型的習慣也可以用最簡單而持久的形式表達出來。系統的設計應該是直覺的，當使用者要做一些他曾經做過的事，系統應可以提醒他該在何處開始。

新介面技術帶來的好處

我們曾討論過新系統很多關於美麗繪圖及哲學上的問題，但究竟使用它有甚麼好處呢？答案很簡單，這些系統易於操作，及有更大生產力，很多人分析這種新系統基本上是爲新手而設，實在是很大的錯誤！

無可否認SMALLTALK SYSTEM及LISP是供專家使用，所以這些系統都曾經作連串的試驗，以找出那種類型互爲作用對使用者最有利。這些試驗，其實兼顧了新手及專家的需要。

因爲專家應用的爭論焦點主要在兩個題目上：從老鼠仔與鍵盤之間轉動並不方便，及新軟件比舊軟件爲慢。

有關老鼠仔的爭論是錯誤的，理由有三個，第一，試驗顯示只有效率很高的打字員利用控制系統可以比老鼠仔所做到的更優勝；第二，系統設有鍵盤控制系統作爲另一個選擇；第三，很多人熟習了新系統後，會發覺用老鼠仔比用鍵盤更順暢。

速度的爭論部份是對，新程式是會比舊的慢，不過，如果程式是小心寫成，速度上可以保持。再者，很多新系統的操作雖然很慢，但在舊的系統上可能是完全沒有的。

經過使用數種電腦系統，去操作程式計劃，已找出使用新介面科技的一套自己體會的理論。當你對某一種系統漸漸得心應手之時，你會愈來愈減少猶豫，發出命令更爲爽快，變成潛意識地處理你的工作。你心裡想，手指便會自覺地按到適當的命令鍵鈕上。

簡單而持久的電腦系統，令使用者的思潮不會被打斷，就好像一個木匠，拿起鎚子，不會有猶豫，它會如何操作。電腦使用者是工程師相信是很少。但新介面科技可令大多數的人在處理日常工作時，變成工程師。

新介面技術的弊端

利用此新科技的程式比舊程式複雜得多。因此，軟件的生產成本自然水漲船高。但不過產品未來前景樂觀，當大量生產後，價格自會回降到合理水平。新軟件所增加的複雜性，刺激發明更具威力的私人電腦語言，如LOGO及SMALL TALK BASIC已不再是這些機器的適合語言，因爲它不能充份利用新機器的繪圖超越能力的優點。

如果舊式的軟件配合新機器的資源使用，它的速度會變得更迅速，新軟件運行速度可與現時8— BIT 軟件媲美。

結論

新介面科技象徵用家——介面設計（USER—INTERFACE DESIGN）新紀元。它的先進生產力潛質最少可與打孔卡（PUNCHED CARD）及整個螢幕編輯器相提並論，可以預料不久將來，每個人在日常生活中都會使用到電腦，蘋果已承諾 LISA 短期內會提供通訊聯繫及資料庫管理操作。這可以顯示新系統將在資料處理及通訊方面担任重要角色。大部分工作，可經聲波指示電腦，消除了鍵盤的障碍。

新介面科技之後，又是甚麼呢？這很趣味性。新科技很可能是關於人工智能，它代表了人類及機器能力最大極限的綜合體。當成爲事實，人類的能力會逐步轉移至電腦上。而結果是一部人工智慧的電腦。□

從指點裝置到觸摸板

流行的視覺指點裝置包括有：老鼠仔（MICE）觸摸板（TOUCH TABLETS），軌球（TRACKBALLS），觸摸螢幕，光筆，數碼板（DIGITIZING TABLETS）及歡樂簽。其中以老鼠仔及觸摸板比其他更爲優越。老鼠仔，一年前，還是「藉藉無名」到現在已是無人不知，不用多作解釋了。觸摸板則已漸漸受人注視。

觸摸板

觸摸板只是嶄露鋒芒的指點裝置，KOALAPAD 觸摸板是其一例子。它是手提式，附有繪圖表面，容許藉手指移動，在視覺顯示器上製造圖案，所接觸的位置轉爲坐標，經遊戲埠（GAME PORT）或數碼據介面（DIGITAL — INTERFACE）傳送入電腦，KOALAPAD 觸摸板爲類似計算機的設計，價格相較之下並不昂貴，數據觸摸墊則較昂貴並可與可以接受RS— 232C輸入的電腦互爲使用。繪圖區域對壓力非常敏感，任何鈍指點裝置例如手指或原子筆均可使用。

觸摸板與數碼板

數碼板是平滑的面，產生準確的坐標位置。用的是手提電力尖筆與觸摸板比較，有缺點亦有優點。數據板在位置顯示更爲準確，觸摸板亦則在文字編輯上較優勝。例如，當打字員須要動手從鍵盤移到操縱指點裝置時，再重新拿起電尖筆；無疑是增添麻煩。

另一方面，數碼板在處理更專門的數碼繪圖時，比觸摸板勝一籌。將手或臂放在板上以穩定持筆繪圖的手指，數碼板不會有偏差，但對觸摸板則很容易造成錯誤。

觸摸板的綫條（保證可以達到大約7 BIT）比數碼板（保證達到11 BIT）亦遜一籌。不過，觸摸板可以選擇一個PIXEL，而能在視覺顯示器。每一空間出現10 BITS。這要靠視覺回饋，用手指在板面劃過。換言之，觸摸板的感應準確性雖是百分之九十九（百分之一錯誤），但亦可以做到準確性重到百分之九十九點九（百分之零點一錯誤）。在最後的分析，觸摸感應適用於大多數的工作中，包括CAD／CAM（電腦輔助設計及生產），視覺回饋是可達到準確的指點。在數據工作上，如地圖或圖表繪製，數據板是較佳選擇。

觸摸板與老鼠仔

老鼠仔是指點裝置，底部有感應器去探測平面上的移動。老鼠仔最値得推薦用於文字編輯上工作，因爲他用是很舒服，同時，除了觸摸版及數碼版外，它的速度亦是最快。

老鼠仔在繪圖上效率比電尖筆及手指遜色。再者，觸摸板比老鼠平宜及佔有較少空間。

不久將來，大多數自動化辦公室文儀生產工具都會有老鼠仔，觸摸板多用於家庭電腦，及作爲商業系統整體一部分。

觸摸板與觸摸螢幕

觸摸螢幕基本上是觸摸板，另外在視覺顯示器前面加上透明物質。觸摸螢幕應用於在公衆地方設置電腦解答詢問，非常寶貴。不過使用者，長時間伸出手臂操作，很易疲累。再者，手指妨碍指點細節事項。（例如文件）由於使用者太接近螢幕，時間會引致流眼水及輻射。

觸摸板與歡樂棒、光筆

歡樂棒及軌球是老鼠仔另一項補充選擇。不過他們指點目的物的速度比老鼠爲慢。雖然歡樂棒是所有視覺指所裝置中，最不昂貴的一種，但缺點是不適用於繪圖及快速指點。

光筆比觸摸版爲廉宜，但它如觸摸螢幕有同樣的弊病。就是用者容易疲倦，亦不會指點細節事項。

概括而言，觸摸版有它眞正的優點，在功能表現及價格上，比其它視覺指點裝置爲勝。

（下轉第29頁）

蘋果最新的操作系統
PRODOS 指令介紹(上)

羅國强

蘋果電腦最新的專業性磁碟操作系操將於短期內面世，它可使 APPLE Ⅱ 採用更高效能的磁碟機及使資料磁碟可用 APPLE SOS (SOPHISTICATED OPERA—TING SYSTEM)直接閱讀。資料從磁碟至記憶體的傳送速度比 DOS 3.3 快八倍。它並可增加時下磁碟的資料及程式儲存容量。

新系統名為 PRODOS 令 APPLE Ⅱ 改良而毋須更改現行利用 DOS 3.3 的 APPLESOFT 程式。

PRODOS 有兩個部分，一是 BOOTABLE 磁碟，有一個檔案名為〝PRODOS〞它儲有代碼處理磁碟操作，而對 RWTS 及 DOS 3.3 的 FILE MANAGER 作出反應。

〝PRODOS〞代碼包括〝機械語言介面〞，有詳細的說明解釋如何將其他編譯語言程式應用 PRODOS 的磁碟操作。而第二個檔案〝BASIC SYSTEM〞負責提供這項能力。

〝BASIC SYSTEM〞使 APPLESOFT 及 PRODOS 連接起來，它供應所有磁碟有關的命令，並令 APPLESOFT 獲得改良增加能力，現在所介紹的 DOS 轉變，就是〝BASIC SYSTEM〞一部份。

DOS 轉變

INIT 不能運作，它的功能已移交到主碟(MAST—ER DISK)工具程式上。如果程式發出 INIT，就會產生 SYNTAX ERROR，並須時刻手頭上都有額外的格式化磁碟，因為當程式運作時，是不能 INITIALIZE 磁碟。

理由是 PRODOS 可容許用上一堆不同的容量的磁碟機，由現時 DISK Ⅱ 至 5—M BYTE PROFILE 硬碟，及其他大細的驅動器，無論是蘋果廠或其他廠所出產的都可以。每一個驅動器都需要其他格式化程式。

在 PRODOS下，新格式磁碟容量沒有〝PRODOS〞的 COPY 或〝BASIC SYSTEM〞佔用了空間。它差不多百分之百可以用作儲存程式或資料，但不能 BOOT 起，如果你想有 BOOTABLE 磁碟，就須要放置〝PROD—OS〞〝BASIC SYSTEM〞及〝START—UP〞檔案在內〝STARTUP〞可以是 BASIC 程式，二進碼式或者是 EXEC 檔案。

因〝BASIC SYSTEM〞關係，連繫上 APPLE—SOFT，它不能認知 INTEGER BASIC，所以 DOS 的 INT 及 FP 命令是沒有功能，反而造成 SYNTAX ERROR。

MON 及 NOMON 在 PRODOS 亦沒有功能，如果程式發出這些命令，將不會收到錯誤的訊息，因為這些命令會被置諸不理。

MAXFILES 亦沒有功能，因為在打開檔案之前，是無須記留檔案緩衝空間，PRODOS 可以全面照顧處理這些功能。如果程式有 MAXFILES 命令，它不能製造出錯誤訊息。但你不能在同一時間開八個檔案。DOS 3.3 則容許十六個同時活躍的檔案。

CATALOG 顯示 80—COLUMN，包括檔案名稱、類別、長度、製作日期、最後處理日期、檔案長度及二進制檔案開始地址等。如果不需要所有這些資料，新的命令 CAT 可以將它濃縮簡短為四十COLUMN。

檔案類別有以下各種：

Applesoft program	BAS
Binary	BIN
Directory(new)	DIR
System file(new)	SYS
Text file	TXT
Variables(new)	VAR
Integer BASIC program	INT

在 PRODOS 磁碟，亦可以有 INTEGER 檔案，不過當 PRODOS 存在時，是無法(RUN)它的。

PATHNAMES(册名)

DOS 3.3 與 PRODOS 一個重要的分別是 PATH NAMES(册名)即是檔案如何編排。磁碟在 DOS 3.3在驅動下有册號(VOLUMN NO.)PRODOS 的格式化磁碟則有名稱，如果你的程式是用册號去分別不同的磁碟，但用 PRODOS 時，就要把號碼改為名稱。

一個〝册〞，是代表整個磁碟，一個 PROFILE 硬碟(相等於三十五個軟碟)是一個〝册〞。每個册均有索引錄，可以最高紀錄五十一個檔案的名稱。如果按下〝CAT〞問册的名稱，PRODOS 就會搜查找尋。這些 DOS 3.3 更為方便。因為 DOS 3.3 要你知道槽(SLOT)的位置，及要看那個索引錄。

〝PATHNAME〞就是册名，是磁碟主要索引錄，除此之外，PRODOS 還有副索引錄(SUB—DIREC—TORY)，假定一個册(一張磁碟)的名稱是〝WORDS〞，作為保存文字處理檔案，內有家庭郵件，發出顧客的函件及雜誌文章。副索引錄可以將不同文件歸納分門別類。

當想看册的名錄，只須打上〝CAT/WORDS〞，PRODOS 就會搜索它在那個磁碟，將索引錄顯現。

PRODOS指令介紹

```
WORDS
  FAMILY  DIR
  CLIENTS DIR
  ARTICLES  DIR
```

DIR 即代表WORDS 的副索引錄，就等於打開一部書，內裡即印上所有在磁碟檔案名稱，首先你會發覺一個表，內容是將册分爲數個部份，每個部份亦有一個表，詳列它的內容——即是名錄。

閱讀家庭郵件的索引錄，須打上“CAT/WORDS/FAMILY”PRODOS首先查 WORDS 的磁碟機，然後是 FAMILY 的副索引錄電腦就會顯示：

```
FAMILY
  MOTHER.01 TXT   6/7/83
  COUSINS   TXT   7/4/83
  MOTHER.02 TXT   10/29/83
```

“/WORDS/FAMILY”構成這個索引錄的PATH NAME 如果決定稍爲更改最後一個字母至 MOM，然後再發送 MOTHER 01，在文字處理程式必須加上 PATH-NAME，“WORDS/FAMILY/MOTHER 02”

另外還有簡便的途徑，無需要每一次都加上整個 PATH NAME 只要你維持每張磁碟的檔案少於五十二個。

PRODOS改變了很多 DOS 3.3 命令的影响，現存的程式不會發覺部份太多的改變，不過在以後的應用上，就會很明顯。

VERIFY只能查看檔案名稱（PATHNAME）是否存在。在 DOS 3.3，這個命令還會查看檔案內的資料。由於大多數程式都是只用 VERIFY來決定檔案是否存在。故此這項功能得以保留。

PR#及IN# 的作用完全改觀。如果你的程式發出 IN#或PR#而沒有〔CTRL-D〕，PRODOS會漠視命令，它不會令指定的槽（SLOT）活躍起來，它不但將自己不連接起來，甚至不發出錯誤的訊號。

如果週邊需要特別的磁碟機程式需要將它與一類東西“掛鈎”，在 DOS 3.3，則要作出特別的程式去提醒 DOS 進行例行操作。

PRODOS提供另一個較簡單方法，以滿足特別例行運作程式的輸出及輸入。 IN#及PR# 現在可以發出功能，以及特別磁碟機運作（SPECIAL DRIVER ROUT-INE） 的地址，如果你的印字機驅動器是以＄300 運行（它將處理資料帶到槽），你就可以按“（〔CTRL-D〕）PR#＄300”，將印字機推動器連接上PRODOS，直至另作指示， PRODOS輸出以＄300 作記憶。

PRODOS 簡單了I-O處理的方法。明確呼喚槽或特別驅動器地址，而令操作系統不會意外地中斷起來。

RUN 亦告改良，PRODOS容許指定行號，在記憶系統內的程式隨時可以有這種能力，你可以 RUN 100 或 RUN 515 去啓動程式執行而無須從頭開始，現在你可以對磁碟上程式同樣如此做，新的行號的參數是“@”替代“DAT”，舉例，啓用執行名爲“BINGO”的程式，行號是 100 ，就打上“RUN BINGO，@100”

RUN 的同組，另有兩個同類的命令。

CHAIN 使你運行連續的程式，而無須失去記憶系統內的變數，以下是例子：

```
1  REM PROGRAM.A
10 A$="THIS is a test"
20 PRINT CHR$(4);"CHAINPROGRAM.B"
```

同時，在磁碟內，你已經保留了：

```
1 REM PROGRAM.B
10 PRINT A$
20 END
```

當 RUN PROGRAM.A，在螢幕上就會出現“TH-IS IS A TEST”當你 LIST 記憶內的程式時，你只會看到 PROGRAM.B.。

你可以 CHAIN 某一指定的行號，一如 RUN

另一個新命令是“-”在任何類型的檔案均可用得。它可以 RUN APPLESOFT 檔案， BRUN 二進制檔案，或者是 EXEC 文件檔案。

PRODOS的EXEC 命令是指引蘋果跟隨磁碟的文件檔案作出指示。作用一如 DOS 3.3，不過還多一點東西，檔案每一段句子是一個欄“FIELD”，而 PRODOS EXEC 命令容許在開始執行之前，“跳過”檔案開首的幾個“FIELD”，只須用新的“F”參數。

PRODOS另兩個新命令是 STORE 及 RESTORE，可以趕走任何的變數或呼喚它回來。

假如在很長的程式上，你會發覺這兩個命令很有用，安排它們當作爲暫停處理的訊號，以作檢查，一個遊戲可以有停頓的選擇，其他的程式可以用它們作爲主要按鈕。你或程式可以 STORE 變數，稍後，你可 LOAD 程式，RE-STORE 變數，及 GOTO行號，而繼續程式。

當發出即時模式，或者是由 CTRL-D（CHR$（4））控制的程式，謹記任何命令都是 PRODOS命令。PR-ODOS 的 STORE 及 RESTORE命令移去所有程式上變數，往來磁碟上VAR 類別檔案， STORE 沒有CTRL

PRODOS 指令介紹

—D控制是 APPLESOFT 命令，只寫出單一指定的陣列去打字出來。它的同伴是 RECALL ，它閱讀陣列內的資料。

新的 PREFIX 命令使 PATHNAMES 運用更自如。

再引用 WORDS 冊名的例子，可以打上〝PREFIX／WORDS／FAMILY〞以代替〝／WORDS／FAMILY／MOTHER.02〞然後再打上〝CAT〞，顯示的索引錄是 FAMILY ONE。以及在該名錄下的檔案名稱，PRODOS 爲任何 PATHNAME 加上 PREFIX，不過前面一定不要有〝／〞。

我們如何知道在那些時間 PREFIX是代表甚麼？只要打〝PREFIX〞，就可以明白，如果想消除 PREFIX，只須打〝PREFIX／〞。

蘋果聲稱每一個採用檔案的程式都有自己的副索引錄，而程式及其有關檔案是唯一的輸入資料，由於副索引錄下可以再有副索引錄，所以這種安排可以收納數以百計的程式，一個程式不需包括長長的名冊去找尋它的檔案，因爲你按下〝PREFIX〞就可搜索到程式的副索引錄。

一張大磁碟內，你可能有很多資料檔案程式都稱爲〝DATA FILE〞，如果每一個程式都有它自己的副索引錄，你就不會在分別資料檔案時所有混淆。

如果需要製造新的副索引錄， PRODOS 的CREATE 命令可以使你如願。此之所以，你可在〝WORDS〞冊索引錄內擁有〝FAMILY〞，〝CLIENTS〞及〝ARTICLES〞副目錄。你可以指示 PRODOS〝CREATE／WORDS／FAMILY〞，或者，如果你較早前已準備好〝PREFIX ／WORDS〞，只須簡單說〝CREATE FAMILY〞。

APPLESOFT 程式內的檔案處理一如正常操作，特別如果你遵照蘋果的指導，將每一組的程式及有關資料都列入副索引錄內，你現存的程式在 PRODOS控制下會正常發揮功能。

下期我們將會繼續講述 PRODOS 的文字檔案、二進碼檔案等的新指令。 ■

（上接第26頁）

老鼠仔將會在未來扮演一重要角式

指點裝置的意義

電腦界正醞釀革命，主要是控制電腦的技巧，用指點視覺符號取代在鍵盤上打出命令。這種稱爲“桌上隱喻”（DESKTOP METAPHOR），在未來將會在商業電腦產生極大的影响。

桌上隱喻是指電腦顯示器上代表文化器材的符號。這些符號並不是死板板被動的圖畫，而是特別設計，在桌上控制電腦刺激活動。假設桌上裝設有電話、檔案文件夾、拍紙簿、廢紙籮等等。這些桌上配件的視覺符號名爲〝ICONS〞。ICONS經電腦作用可刺激它代表的眞實物體，當使用者用視覺指點裝置（老鼠仔）指點它們時，ICONS便變得活躍起來，進而操作一項電腦工作。例如，指點一叠紙張的圖畫及廢紙簍圖畫，即代表你想將該叠紙張拋掉（或從螢幕上移去）。

一般而言，你可創造，輸送，編檔案，及取消訊息，圖表，電子部件及其他文件，只須簡單指點ICONS。電腦使用者無須再是專家，才能處理這些工作。使用者只須靠圖型及 ICON 的特別編排去提供淸晰及直覺功能提示。

ICON 的直覺本質是桌上隱喻的中心點。使用者對眞實世界的桌上文儀器材的知識，可以在電腦 ICON 符號世界中進行工作。

桌上隱喻其實只是普遍性原則“物質隱喻”（PHYSICAL METAPHOR）其中一個特別個案。“物質隱喻”是由ALAN KAY 所提出來。他的概念是如果電腦程式能夠像以物體，而非用數字抽象符號去運作，那麼電腦就更容易學習使用。人們通常對物體有較佳直覺。這種直覺可以幫助他們明白電腦。若電腦呈現的是類似物體。

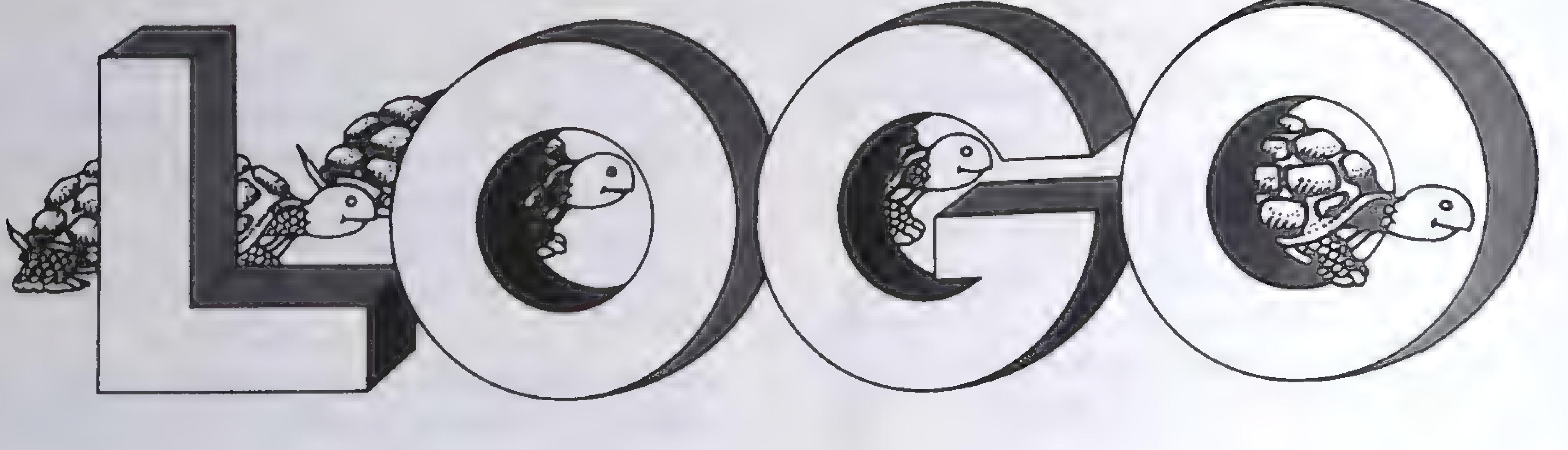
在這些系統，使用者可以找出用符號顯示的重叠紙張，不斷移去在上面的紙張，可減少受視綫阻礙而不能看到的紙張。這令儲存及查閱檔案更加完整簡單，而毋須打上〝LOAD FILENAME EXT〞。

“物質隱喻”所顯示的人旨機器溝通的程度可能並不明顯。爲了解釋隱喻的好處，試想，用鍵盤打字方法去控制電腦，涉及多種不同腦部技巧及腦的不同部分。但用桌上隱喻，只須知道圖畫的意義及指點ICONS。

視覺思考在腦的右邊進行，而語言思考在腦的左邊進行。視覺及語言溝通時，基本上是左邊的腦。這亦是採用鍵盤方法時，左腦亦在動腦筋。腦的右邊，在另方面，則涉及更多關於視覺思考，並相信是藝術家及思想家創作的源頭。事實上，更重要是視覺資料比語言資料或文件資料更易記牢。 □

不是小孩兒的玩意

被人誤解的LOGO語言



LOGO 這種電腦語言是一位名叫西摩亞·巴帕（S—EYMOUR PAPERT）的大學教授，與幾個助手於十六年前共同發展出來的。亦由於西摩亞教授寫了一部名為"M—INDSTORMS—CHILDREN，COMPUTERS AND POWERFUL IDEAS"，使到 LOGO 普遍地給人知道。這部書立論認為 LOGO 能令到兒童在他們的普通教育中，得到適當的電腦智識。

但亦由於這部書，有些人便以為 LOGO 只是給兒童使用的玩具語言，只在小學通行。可是他們却不知道，在愛丁堡（EDINDURGH）大學的研究者，在人工智能的研究上，已應用了 LOGO 語言呢。

LOGO 有時被指為屬於 LISP 的子集。雖然在某些地方它確是從 LISP 中改良過來的，但它的形式却是更加簡潔和程式的可讀性更優良。

對 LISP 來說，它主要的構思是所有資料都可以"原子"（ATOMS）和"列表"（LISTS）來代表。列表可以是字組的列表、數字的列表、字母的列表，或列表的列表。事實上你可以有列表的列表！由於首部能使用 LISP 的電腦的部份原因， LISP 列表的第一個份子便叫作為CAR，其餘的被叫作為CDR。

利用 CAR 和 CDR ，列表便可以被切開、檢視、分類、混合、洗去與重新安排。 CAR 送回列表的第一個字組； CDR 就送回其餘的。在 LOGO 亦有這些功能，不過它們被叫作 FIRST和 BUTFIRST。

若你認為筆者在賣弄太多的 LOGO 語言的列表處理功能，我告訴你我只不過是不想你們說：「那麼 LOGO 豈不是真的 LISP 的附屬品？」。因為若讀者有聽聞過 LOGO 語言的話，多數是與「烏龜繪圖」（TURTLE GRAPHICS）有關的。烏龜是西摩亞教授給機械人的別名。他在 LOGO 的發展過程中，其中一個主要的貢獻便是從他的一個機械人身上，得到靈感的。而這個貢獻便是：教授認為兒童需要一道概念化的橋來溝通他們在物質化的世界中身受的經驗和一個程式的形式化與及抽象世界。而那個機械人代表了一個強烈的突破。西摩亞教授再使兒童們像命令機械人般，互相命令對方去依著同一的幾何圖形行走，從而令到兒童領略到怎麼去編寫程式。這個強有力的構思亦因而被採用到其它方面去。

『"烏龜繪圖"能夠被其它電腦語言採用』。若這說法就是關於 LOGO 的話，那麼它早就被遺忘了。幸好CP／M操作系統的編寫者，嘉利·基戴爾（GARY KILD—AVE），亦是DIGITAL RESEARCH 的總裁，他自己在 IBM 私人電腦貫切了 LOGO 的使用，給了人們兩個啓示： LOGO 語言在將來是一個理想的應用程式語言；我們應小心去留意它，

被人漠視的原因

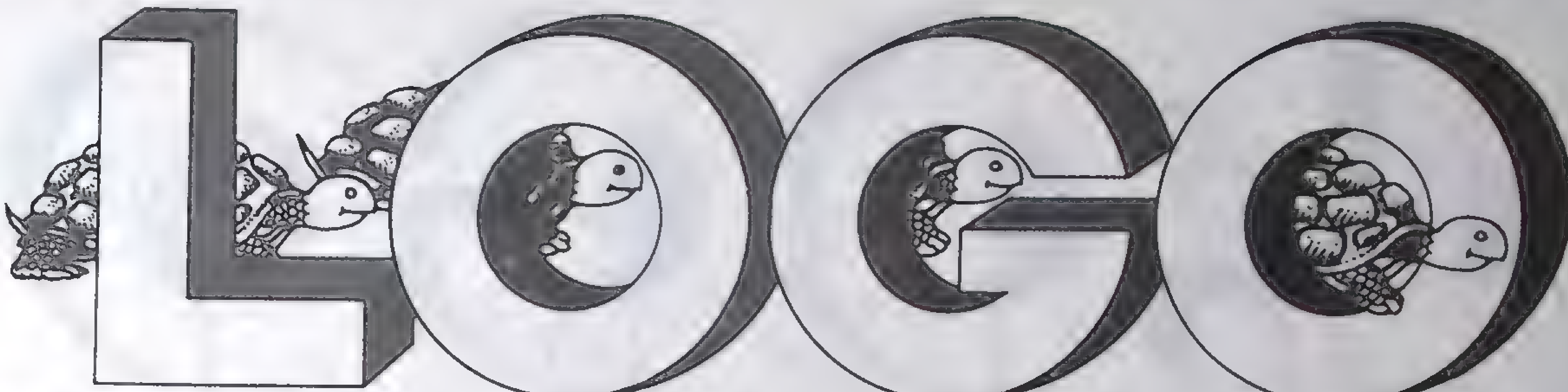
那為什麼 LOGO 在以前不能被熱衷地作為一種普及的電腦語言呢？在衆多的原因當中，只有兩個是好的理由。一個就是 LOGO 很消耗記憶貯存，它的列表處理操作會積聚資料於記憶貯存中，因此很快令到可用的記憶空間滿溢。另一個有關的原因是當要實行LOGO 時，就算是將它與實行一個好的BASIC來比較，它的過程多數較緩慢。

其它不好的理由有：㈠1970年代的 LOGO 用家們及職業教育者沒有特別地去推廣 LOGO ；㈡電腦用家及製造商都有一個觀點，他們認為 LOGO 是一種智力性的LEGO 罷了！

這些不好的理由，現已成為過去了。一九八二年八月的 BYTE 雜誌，介紹 LOGO 給成年人電腦用家；而 LOGO COMPUTER SYSTEMS INC（LCSI）於 MONTR—EAL 成立，供給這種語言商業性的推動。現在亦有大量優良的著作和論文去糾正早期對於 LOGO 的錯誤概念。

至於在編寫一個有效率的 LOGO 翻譯程式的技術，在過去這三年間已有極大的改進。而APPLE是不容許它在這方面的落後的。因為假如DR LOGO（是對DIGITIAL RESEARCH 所提供的稱讚式叫法）在所選擇的市場成功的話，那麼到時將會有強而有力的 LOGO 供 LISA 採用。

另一個發展似乎更可推前 LOGO 的成立。現在很多數



學上的基本功能，例如加、減、乘、除等等，都可以由硬件去處理，同時新的電子元件更可處理更加多的複雜的數學功能，故怎會沒可能去發展一些特別的微處理器去照顧LISP和 LOGO 的列表處理功能呢！

現時在美國已有些特別建造的 LISP 電腦，它們被用作去發展 LCSI 對 LOGO 的實行。無論怎樣，一部L—ISP 電腦仍然需要六個數字的價錢，但這總會改變的，硬件將可被採用來製作迅速和經濟的 LOGO ，而且是効能強勁和容易學習的。

誰學LOGO

此刻我們應該談談什麼人應該學 LOGO 呢？若你想引導你的孩子或祖父母去認識何謂程式，那麼你可以考慮L—OGO，使它成爲 BASIC或 FORTH 之後 另一個優良的代替。雖然他們未必可以去編寫一個普及分類賬項軟件，但要令他們明白程式編寫究竟是什麼一回事，那麼 LOGO 正是一處適當的起步點了。

同樣的，若你正利用 APPLE 在文字處理、部記、和 VISICALC 方面應用，而想嘗試去編寫程式的話，那麼請選擇 LOGO吧。它不用費太多的時間便可以掌握得到，更好的是它可以給你樂趣呢！

有興趣在人工智能研究方面，但又想避免了要去學習LISP的麻煩的話，相信 LOGO 定可以滿足他們的。

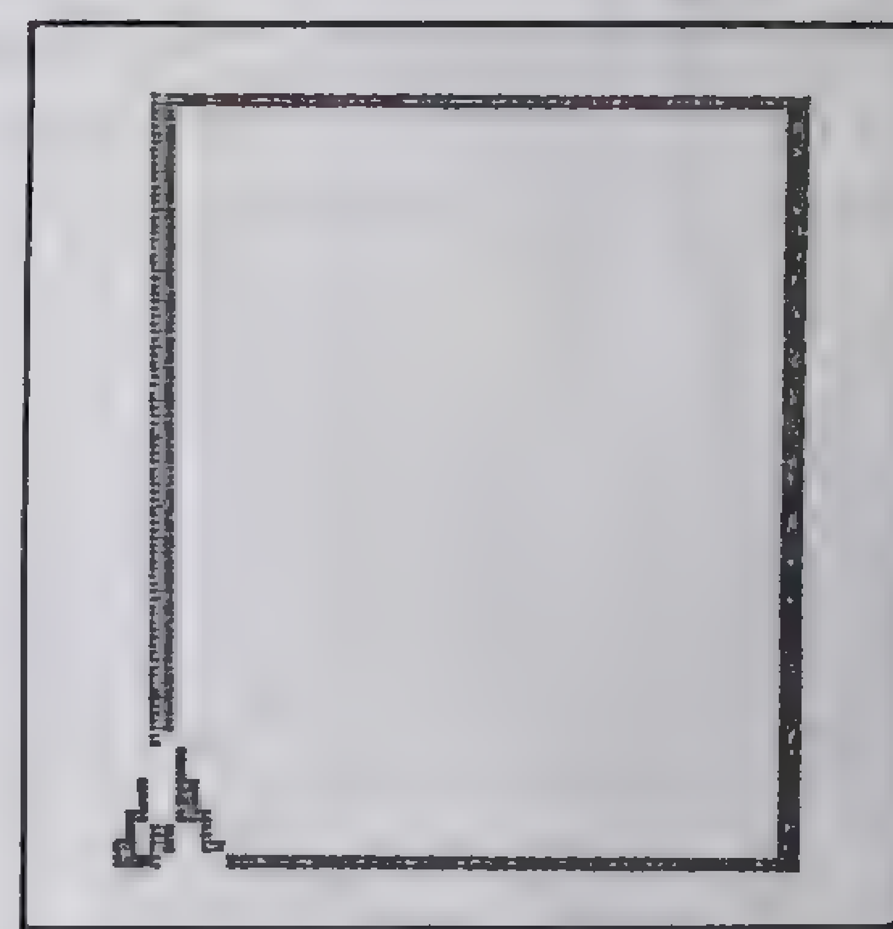
試繪正方形

以下是介紹如何利用 LOGO 幫助你在螢光幕上繪圖。LOGO 的初學者通常首先學習繪劃一個正方形的程式，這可透過三種不同程式做到。第一種程式爲：

```
TO SQUARE
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
END
```

這是最簡單，但最笨拙的方法，而只可以繪劃同一大小的正方形，換句話說，當輸入這一套指示之後，電腦已將正方形之形像固定，如果在鍵盤上按 SQUARE，螢幕就會出現圖（1）的圖形。正方形的左下角蹲伏一着小龜，在這套指示上，RIGHT、FORWARD 與及 LEFT、BACK—WARD 全都是 LOGO 的基本字彙。

圖(1)



至於第二種程式，則較爲普遍：

```
TO SQUARE :SIDE
REPEAT 4 [FORWARD :SIDE RIGHT 90]
HIDETURTLE
END
```

這個程式較首項爲精簡，避免了重複 RIGHT & FORWARD 多次，而且可以隨意繪劃不同大小的圖形。例如，當按入指示 SQUARE 50 之後，螢幕便出現一個如圖（1）的正方形，但若指示爲 SQUARE 25，出現的便是（1）的正方形，但若指示爲 SQUARE 25，出現的便是面積爲圖（1）中四方形四分之一大面積的正方形。換句話說，圖形的面積就可隨着指示而改變。當繪劃工作完畢後，小龜便會自動失踪。

（HIDETURTLE、SHOWTURTLE 與 PENUP、PENDOWN 均是 LOGO 的基本字彙。這些字彙就是 LO—GO 語言結構的一部份）。

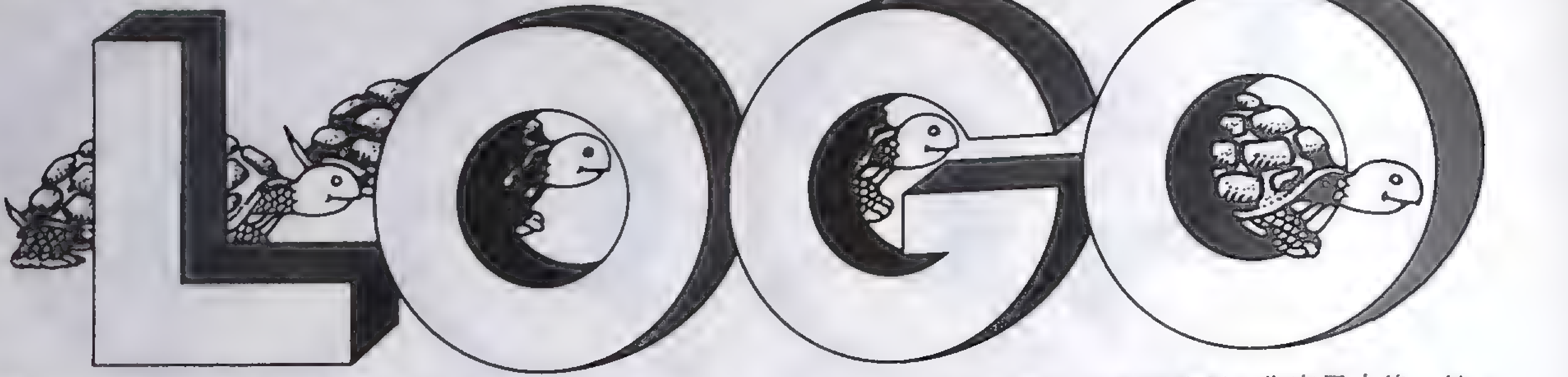
第三種辦法其實就是最爲普遍使用，雖然驟眼看來似乎甚爲複雜，但其優點是十分之多的：

```
TO SQUARE :SIDE :ANGLE :ANGLESUM
IF :ANGLESUM > 360 [HIDETURTLE STOP]
FORWARD :SIDE
RIGHT :ANGLE
SQUARE :SIDE :ANGLE :ANGLESUM + ANGLE
HIDETURTLE
END
```

當輸入程式之後，如果按指示 SQUARE 50900 ，螢幕上就會顯示出圖（1）之正方形。它的優點就在於其折返性，確保小龜在繪劃完畢後回到原本位置，然後消失。

剛剛提及到的程式，除了劃出正方形之外，還可輸出任

被人誤解的LOGO 語言



何等邊圖形，所需要的轉變就是替他換上較適當的名稱——TO POLY：SIDE：ANGLE：ANGLESUM。當使用這程式時，須謹記： ANGLESUM的價值等於0，而其他的轉變份子則任由選擇，圖（3）的三角形，正方形及五角形可同時在螢幕上顯出，當然你首先就要按下指示：

```
TO POLYGONS
PENUP SETPOS [-120 0] PENDOWN
POLY 20 120 0
MOVETURTLE
POLY 20 90 0
MOVETURTLE
POLY 25 72 0
END
```

這一則指示中，MOVETURTLE首先要輸入定義的：

```
TO MOVETURTLE
PENUP
SETX XCOR + 50
SETHEADING 0
PENDOWN
END
```

SETX及 SETY 可以指示小龜移動往螢光幕上特定的座標。至於 XCOR或者 Y COR則領導小龜回到經常之位置。而 SETX X COR＋50 的結果就是將小龜在螢幕板上向右移 50。PENDOWN 時，小龜在螢光板移動會留下綫條，而 PENUP 則相反。

SETHEADING指示小龜的方向，而0是向北，意即指向幕光板的正上方，90即東，180即南及 270 即西，餘此類推。

看過上述的解釋之後，聰明的讀者或許已經知道如何繪劃一個正多邊形，如果要劃一個N邊形，只要按下列的指示便可：

POLY：SIDE 360／N 0

換句話說， POLY 50 360／13 0會繪劃一個正等邊三角形，而 POLY 1 360／360 0 則會出現一個圓形。明白已上所述，你對 LOGO 的大概原理，應該有一個初步的認識。

DRAW

DRAW 及 LOGO 中最基本但又最重要的程式。使用 DRAW 時，小龜在螢光板上位置可隨意移動，只需在鍵盤上按下單字母。（U即向上，D即向下，L代表向左，R即向右）當小龜到達理想的位置時，你只須按下另一個單字母，螢光板上便會現出指示的圖案。（S代表太陽、A代表大樹、B即小鳥、而W則是風車）當然這些單字母的意思一早已經在這程式中定義。而所有繪劃工作完畢之後，按下0（或者Q）之後，便可使用其他的程式。

在附列的程式表上，有 READCHAR 一字，這是LO—GO 的基本字彙，意即等待下個單字母在指示。〔 〕是BASIC的THEN意思。

DRAW 這程式內，只有一個變值（：SIDE），故此可以繪劃不同大小的圖案。在操作時，：SIDE 的價值會顯示在螢光板上的左下角，按下單字母便可改變圖形大小，令你的圖畫上有不同大小的樹木及雀鳥，甚至太陽爸爸、太陽兒子，視乎你的喜好。（H即縮細一半、T則三分一、Z則放大一倍，3則倍三）

現在再需要解釋的是如何註定圖畫上各不同的份子（詳細的指令見於附表）：

SUN 利用 POLY：ANGLE：COUNT，令到現出一個有光綫伸出的圓形。

BIRD 的 ARCL 及 ARCR，都不是 LOGO 的基本字彙，但都是內置在 APPLE Ⅱ LOGO SYSTEM。

WINDMILL 比較前二者爲複雜，分爲三個部份，即 DRAWBODY，DRAWROOF及DRAWSAIL，而它們是從序操作，即是當 DRAW 時，按下單字母W，從螢光板上你可以看到首先出現風車的屋身，跟着屋頂，然後風車葉，組合而成一個完整的風車。而三個部份當然合比例，因爲使用同一個：SIDE。

TREE 是附表上最複雜的圖形，當按下「T」時，首先便會出現樹幹，這裡是採用隨意式繪劃，令到樹幹不致四四方方，增加自然感。跟着便到 TREETOP，TREETO—P 令到另外兩部份進行，即 LSIDE & RSIDE，即樹頂之左右部份，兩者不同地方只在於使用 LBRANCH及 RBRANCH。

其實使用這個程式並不及加上自己的圖案有趣，你可以自己設計一些門、窗、烟通及車房加入，繪畫你的居室，又或者加入船、波浪、海鷗及燈塔等物，設計一幅海景。這個程式除了可以教導幼孩子一些景物的名稱（按字出圖），另外又可以幫助你花園、辦公室的平面設計。

你使用 DRAW 時，除了依照上述指示外，又可加入一些變化。你可將 PENUP 轉爲 DOWN，那麼當小龜在螢光板上移動，便會留下綫條。又或者當 PENUP 時，你又可以指示小龜移動的速度增快，其法如下：

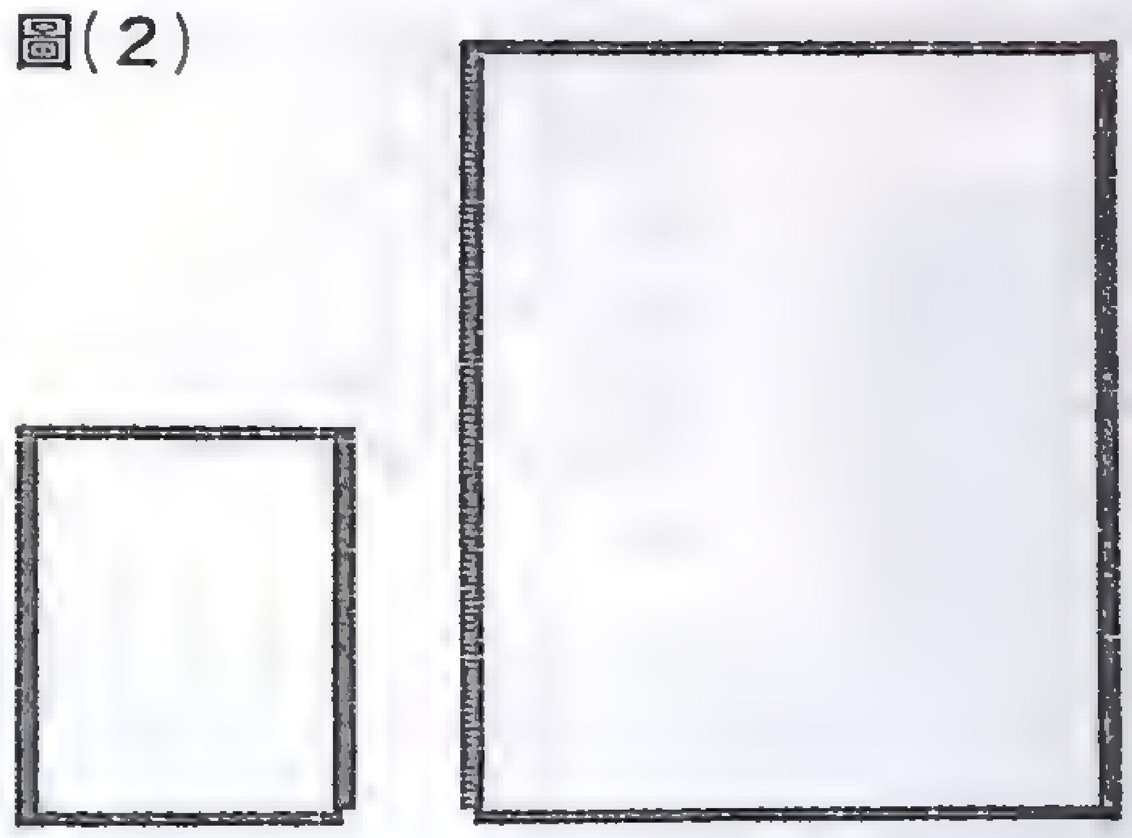
IF：K=″F〔FAST〕

然後註定 FAST，令小龜移動的幅度擴大至10或20當按下U、D、L & R 等指示：

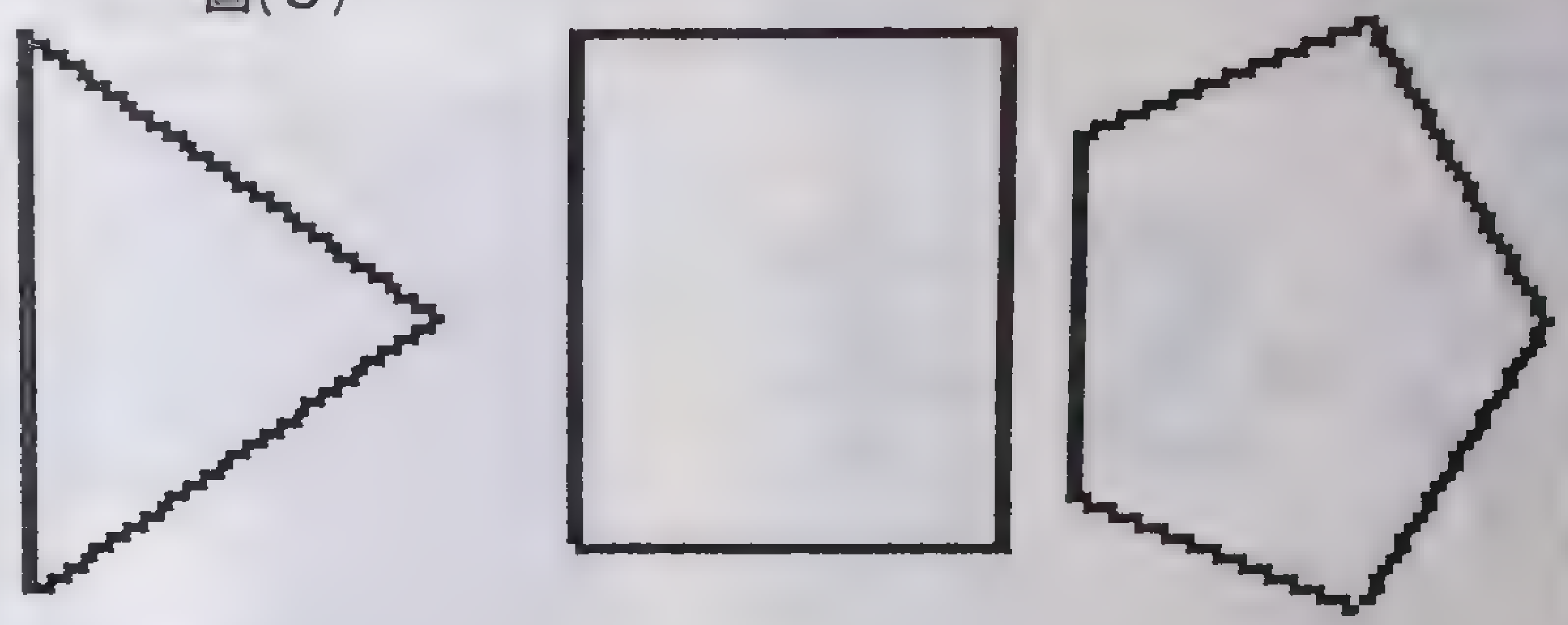
IF：K=″L〔SETX X COR—10〕

當按下F及L之後，小龜就會向左移10，而不是移1。

圖(2)



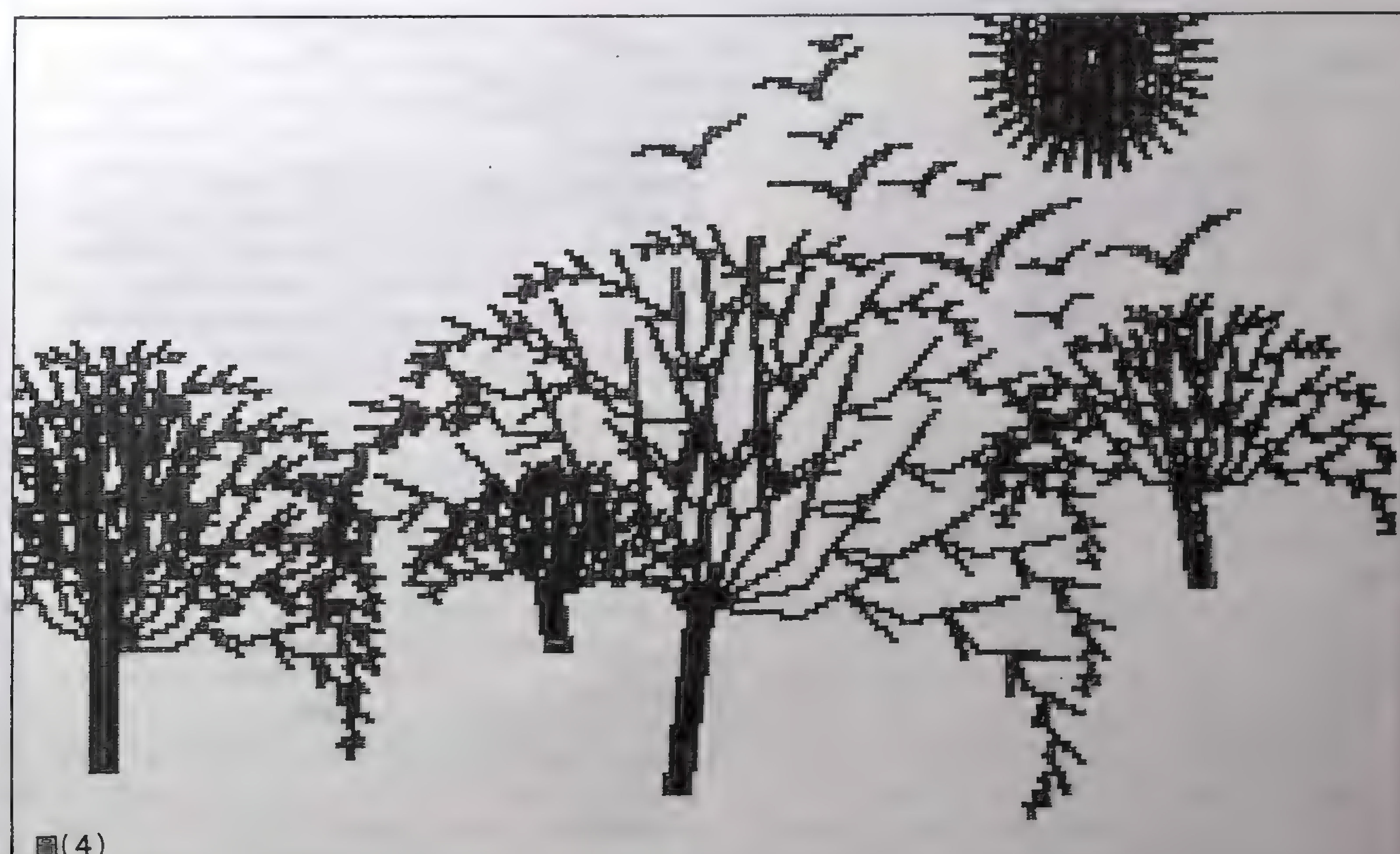
圖(3)



假如你擁有一具彩色的螢光板，當然希望加進適當的程式令到畫面更多姿多彩，不過要留意的是輸入新的程式時，有可能將螢光板上根本有的圖案洗去，在這個時候，你需要一部 SAVEPICTURE器具，這可令到記憶系統能夠裝載更多，這 SAVEPICTURE在市面上應該可購買到，但並不屬於 APPLE II LOGO PACKAGE 的附件，不過如果你買的是 TERRAPIN LOGO ，就附有此器具。

利用印字機印出圖畫的過程，不同的組合，有不同的方法。例如 LCSI APPLE II LOGO，你要利用 UTILITY PROGRAM，從外面將圖畫當成普通 BINARY FILE 儲入。

對於我本人來說，我覺得 LOGO 的圖畫太過幾何化，所以我時將輸入一些隨意的指示。而 LOGO 吸引人的地方是它深淺的幅度十分大，由幾歲的小孩童，至到要加入精深的數學程式都可。如果你對 LOGO 十分感興趣，以下一本書可介紹給你參考：「DISCOVERING APPLE LOGO：ANINVITATION TO THE ART AND PATTERNS OF NATURES，作者是 DAVID THORNBURG ，由 ADDISON WESLEY 印行，希望你利用 LOGO 畫出理想的圖畫。□



圖(4)

```
?PO "DRAWSCAPE
TO DRAWSCAPE :S
WINDOW
DRAW :S
END


?PO "DRAW
TO DRAW :S
MAKE "K READCHAR
IF :K = "H [MAKE "S :S /
   2]
IF :K = "T [MAKE "S :S /
   3]
IF :K = "2 [MAKE "S :S *
   2]
IF :K = "3 [MAKE "S :S *
   3]
IF :K = "L [PU SETX XCOR
   - 1]
IF :K = "R [PU SETX XCOR
   + 1]
IF :K = "U [PU SETY YCOR
   + 1]
IF :K = "D [PU SETY YCOR
   - 1]
IF  :K  = "O  [HT  THROW
   "TOPLEVEL]
IF :K = "S [PD SUN :S *
   2]
IF :K = "A [PD TREE :S]
IF :K = "W [PD  WINDMILL
   :S]
IF :K = "B [BIRD :S / 2]
PRINT :S
DRAW :S
END


?PO "SUN
TO SUN :S
POLY :S 170 0
PU HOME
END


?PO "BIRD
TO BIRD :S
PD
LT 45
ARCL :S 65
RT 180
ARCR :S 65
LT 120
ARCR :S 65
RT 180
ARCL :S 65
SETH 0
END


?PO "WINDMILL
TO WINDMILL :S
PD
DRAWBODY :S
DRAWROOF :S
RT RANDOM 90 LT 15
REPEAT 4 [DRAWSAIL :S LT
   90]
END


?PO "DRAWBODY
TO DRAWBODY :S
RT 90
FORWARD :S
RT 90
FD :S * 6
RT 90
FD :S * 2
RT 90
FORWARD :S * 6
RT 90
FD :S
END


?PO "DRAWROOF
TO DRAWROOF :S
FD :S
LT 120
FD :S * 2
LT 120
FD :S * 2
LT 120
FD :S
END


?PO "DRAWSAIL
TO DRAWSAIL :S
FD :S * 6
LT 90
FD :S
LT 90
FD :S * 5
LT 90
FD :S
LT 90
BK :S
END


?PO "TREE
TO TREE :S
SETH RANDOM 20 LT 10
MAKE "DIRECTION HEADING
MAKE "SAVEPOS POS
TRUNK :S * .8
SETH :DIRECTION
FD 2 * :S * .8
TREETOP :S * .8
END


?PO "LBRANCH
TO LBRANCH :SIZE :TRUNK
IF :SIZE < 2 [PU SETPOS
   :SAVEPOS SETH
   :DIRECTION FD :TRUNK
   PD STOP]
FD :SIZE
RT 40
FD :SIZE * 1.5
BK :SIZE
LT 80
FD :SIZE * .75
BK :SIZE * .75
RT 20
LBRANCH :SIZE * .75
   :TRUNK
END


?PO "TRUNK
TO TRUNK :SIZE
FD :SIZE * 2 LT 90 FD 2
   LT 90 FD :SIZE * 2
   RT 90 FD 2 RT 90
   FD :SIZE *2
END


?PO "TREETOP
TO TREETOP :S
MAKE "TRUNK :S * 2
MAKE "ANG1 90
LSIDE :S :TRUNK :ANG1
RSIDE :S :TRUNK :ANG1
END


?PO "LSIDE
TO LSIDE :S :T :ANG1
IF :ANG1 < -20 [STOP]
LT :ANG1
LBRANCH :S :T
LSIDE :S :T :ANG1 - 20
END


?PO "RSIDE
TO RSIDE :S :T :ANG1
 IF :ANG1 < -20 [STOP]
RT :ANG1
RBRANCH :S :T
RSIDE :S :T :ANG1 -20
END


?PO :RBRANCH
TO RBRANCH :SIZE :TRUNK
IF :SIZE < 2 [PU SETPOS
   :SAVEPOS SETH
   :DIRECTION FD :TRUNK
   PD STOP]
FD :SIZE
LT 40
FD :SIZE * 1.5
BK :SIZE
RT 80
FD :SIZE * .75
BK :SIZE * .75
LT 20
RBRANCH :SIZE * .75
   :TRUNK
END
```

暢銷大榜上的寵兒

Lotus 首張綜合性軟件



1-2-3

在美國暢銷軟件大榜上目前風頭最勁的應該是 LOTUS 的1—2—3 合成軟件了。

1—2—3 是一張集所有管理策劃系統於一身的磁碟，它包容了過去只能在個別軟件才能找到的功用。 1—2—3 的製造商 LOTUS 宣稱這是一項大躍進，將一名管理人所需要的東西全部集中在一起。

一如傳統的 SPREADSHEET 程式，1—2—3 的計算紙（WORKSHEET）是由許多行與欄的單元（CE—LL）組合而成。每個單元儲存一項資料——數目、方程式或標誌。而螢幕所顯示的只是一張計算紙的部份。

1—2—3 有多個版本適合 IBM XT、DEC、SI—RIUS、WANG PC 電腦，不久將來，連最新的 MA—CINTOSH 版本也會面世。

1—2—3 綜合了幾個不同的財經操作系統，但毋須要用有經驗的會計員亦能操作自如。這些系統包括有：財經策劃、商業繪圖及數據管理。對於管理階層用家來說，1—2—3 是繼 VISICALC 之後的一張偉大的軟件。

VISICALC 與 1—2—3 最基本的差異是前者較難去駕御，而 1—2—3 則非常容易去使用。不過，無可否認 VISICALC 是一件非常出色的作品，甚至連設計 1—2—3 的原作者 MITCH KAPOR 也承認，1—2—3 是根據 VISICALC 爲藍本而發展出的軟件。他原意是編寫一個綜合圖案和類似有 VISICALC 或 SPREADS—HEET 的軟件。

當然，1—2—3 亦有本身的缺點，它佔用許多的記憶，要充份發揮這件軟件的優越性能，就必須要最少 256KB 的記憶。在某些情形，還需要要更大的記憶量。不過，對於大部份的操作，一般記憶量的機器也能勉強應付。另一點是其文字處理部份，應該可以做得更好。不過 LOTUS 亦表示會於今年稍後推出 ENHANCE 版本。

LOTUS 1—2—3 的售價約爲港幣四千餘元左右。略嫌稍貴。

附同磁碟一併發售的兩本參考指南及一本使用入門手冊。

對於新使用者，LOTUS 還備有一張〝1—2—3 ELECTRONIC TUTORIAL〞磁碟，要初步認識1—2—3 的功能實在非常有幫助。

當 BOOT 起 SYSTEM DISC 後，鍵入 KEY—BUK〔RETURN〕及 TUTOR〔RETURN〕，這張 TUTORIAL 磁碟會有如下的顯示：

A Getting Started
B The Loan-Analysis Worksheet . . . I
C The Loan-Analysis Worksheet . . . II
D The Loan-Analysis Worksheet . . . III
E Handling a Database
F Graphing

從這索引當中，可窺見這個軟件的多項功能。

由於 1—2—3 是來自 VISICALC，故許多指令及螢幕顯示模式都是似曾相識。它較 VISICALC 出色的地方包括：VISICALC 的欄濶並沒有足夠的選擇。1—2—3 則改良了這項缺點。視乎記憶容量的多少，1—2—3 的計算只可劃分成 256 欄乘 2084行。1—2—3 載入及反應時間較 VISICALC 快兩倍。

總的來說，1—2—3 應用軟件的優點相當令人滿意。

電腦科技的搖籃——矽谷

工作至上的地方■婚姻生活的墳墓

有人把矽谷描繪成美國工業發展後對未來憧憬的體現，是自由企業的縮影，是一個任何有頭腦，有創意和勇氣的人只要肯努力善加利用，就可在電子業中一夜致富的地方。

可是，精神治療家和警務人員却說，有越來越多的跡象顯示，矽谷這個具有高度壓力的競爭環境，其離婚、虐待兒童、酗酒和吸毒的比率均有偏高的迹象。

巴羅艾託鎮的一位專治精神病的專家佳娜・克瑞斯博士說，「這是今日的淘金熱。有一種隨時可以突然致富的現象使人們覺得『我也能像別人一樣獲得成功』，『某人眞的發了大財，我必須拼命努力趕上』。你非常担心只要一停止工作別人就會馬上超過你。」

克瑞斯有一個病人在事業上非常成功，但是婚姻觸礁了。這個病人說出了自己的狀況：

你眼看四週的人都賺了好幾百萬，你就會想他又不比自己聰明，在技術上也不比自己懂得多，要作他所作的事情可說根本輕而易舉。但是如果你現在不做，十年之內大概都不可能再碰上這種機會了。你不能放鬆自己，因爲每個星期一早上就在你喝咖啡的時候，都會在報上看到另一個成功的故事。你唯一的辦法只有每週工作八十小時，這樣一來你太太當然產生反感，家庭因此而告破裂。」

正如一位工程師所說，這種「掠奪性的喀爾文教派特質」主宰了加洲北部的矽谷，在這裡小小的矽半導體晶片建立了龐大的電腦財富。這裡是上千家高級科技公司的大本營，是全美各行各業領導人士所羨慕的地方。許多一等的技術人才，受到勤奮致富的報導所吸引而移居的矽谷。

可是根據洛薩托斯一個非營利研究機構柯諾斯公司的裘蒂・拉森以及凱洛爾・吉爾兩人的研究報告，這種期望經常落空。

他們在最近一項名爲「改變中的矽谷生活方式」的研究中寫道：「矽谷的神話是個夢想。」

他們還指出：「夢想成眞的成功者眞是鳳毛麟角。矽谷的失敗者跟成功者可能成十與一之比。長期的工作和不斷的壓力，已使許多人精神不支。也許影响最深是人際關係的惡化。」

許多公司已開始注意到這些問題。一些較大的公司如美國半導體公司以及惠普公司，現在都在工廠附近設置精巧的娛樂設備。甚至一些小規模的新公司，也都至少提供淋浴，好讓員工利用午餐時間慢跑以消除緊張，山景市和毗鄰社區內的街道上，慢跑者多如江之鯽。

在矽谷，工作主宰了一切。秘書們說，看別人孜孜矻矻忙到晚上，而自己却準時下班會有一種罪惡感。史丹福大學一個編緝囘想最近一次晚宴的情形說，那些男人三句不離本行，除了電腦不談別的。她說：「一般男人是透過足球賽來認識對方，在這裡則是透過電腦。」

當然，在別的工業中，壓力問題也普遍存在。但是在矽谷有許多人認爲，由於驚人快速的創新，每項計劃都迫切需要趕快完成，因此壓力特別大。只要有個對手早幾個月成功，多年的努力便可能付諸東流。任職一家大公司的一個女性高級工程師說：「每日的工作都息息相關。有時我對週末不來上班會覺得內疚，但是我需要休息。」

佔矽谷大部份地區的聖克拉拉郡的離婚率是全加州最高的。據加州衞生部表示，由最近的資料獲知，一九八一年當地的離婚率高達千份之七點一，而加洲只有五點八，全美國只有五點三。

當地地方法院檢察官助理羅伯特・馬斯特遜，聖克拉拉郡的性干援和虐待兒童案例比率「極高」。他認爲這些問題跟工作的壓力，以及家庭不能提供正常情緒滿足有關。他說，另一個副作用是，兒童吸毒的比率甚高。

克瑞斯博士說，許多夫婦過着一種入不敷出的生活。一個工程師可能有年薪六萬美元的收入，但是他跟太太兩人基於反正很快就會很有錢的想法，花起錢來毫無節制。克瑞斯博士說：「這種人都是舉債渡日，他們可以一個晚上揮霍五百美元，飲用最高級的酒，買最高級的衣服，而同時他們又像狗一樣的拼鬥。」■

日本電腦的侵略

美國微型電腦界悲觀人士深信本土的製造商不久將會被外來「侵略者」所埋葬。而英國的微型電腦的企業也面臨地震。

他們的憂慮是可以理解的，美國的汽車業、錄影帶及電單車工業、攝影機及音响業曾經叱咤風雲，執世界牛耳，但近年受到本港的強烈競爭，差不多一蹶不振。

美國的電腦業是否會受到同一命運呢？但照目前的形勢分析，情況還沒有達到這個地步，日本的電腦「侵略」還是遙遙無期。

一位在日本工作多年的電腦雜誌專欄作家表示：英國缺少日本微型硬件的情況，肯定即將改變……遲早水壩大閘將會大開，此段文字說於一年前，但到目前爲止，我們仍等待着水閘滑開。

日本電腦輸出阻滯有幾個明顯原因。軟件促銷硬件：但日本所寫的軟件（日本文寫成）並未能在歐美成爲最暢銷的產品。

不過語言的障阻，並不能夠完全解釋日本電腦滲透海外市場不成功的原因，沒有人可以阻止日本公司獲得准許證購買 CP／M及 MSDOS。事實上，他們已經如此做，大多數微型電腦商業機構主要是銷售標準軟件，而日本硬件製造商亦能利用 WORDSTAR 及 SUPERCALC 去配合，而他們亦做得出色。日本長久以來，被公認爲廉價抄襲複製家，但近年已漸漸洗脫「污名」，它所產生的高科技產品性能，品質佳已贏得認同。音响設備市場，日本公司競爭力強。世界任何地區上所產生的微型產品，差不多都有日本的晶片在內。

在繪圖方面，日本公司亦領導全球，三種現成編寫的語言製造了日本公司成功因素。

日本產品的良好品質管制，彩色斑爛奪目的圖案，使用標準軟件包裝，以及給百份之五佣金予經銷商的優待等等，應該是有利的推銷條件，但爲何日本微型電腦還未能在出口市場上爭取到優勢呢？

原因是很微妙而暗晦的：此與市場學、產品生產綫的選擇，日本電腦行業及日本人的性格有關。

發行出漏洞

開拓市場上，日本公司走錯了路，資訊科技研究顧問公司總裁賈拉咸曾作以下分析：「日本商進軍美國市場初期，介紹新產品時，已鑄成很傳統的市場學錯誤。」問題繫於發行方面。

NEC，日本微型電腦市場上的領導性公司，兩年前在英國推出PC—8000系統，而選擇 IBR 爲經銷商。 IBR 以代理電子零件聞名，而經管微型電腦還是首次。一年後，由於推銷沒有成果，NEC 終於取消了 IBR 的代理權，而改由自己做經銷。 SORD 則選擇 EXLEIGH BUSINESS MACHINES 爲經銷商，成績欠佳，現在亦是在倫敦自設辦公室負責經銷。日立的 16000 與 IBR 可以互爲使用的新產品，九個月前已在倫敦宣傳，但至今仍未上市，阻延的原因，是未能與零售商在售價上獲得協議。而支援連鎖機構設立亦緩慢。

日本 A1 電子公司亦遭遇同樣問題，自行經銷又出阻滯，現在又要重新選覓新經銷商。

日本大公司所受到的困擾，並非是單一個別性問題，東芝， FUJITSU，MATSUSHITA，PANASONIC 亦同樣觸礁，前兩間公司已經重新檢討銷售策略。PANASONIC 的廣告字眼已改爲我們知道產品昂貴，但請細看品質，是物超所值。

訂價一場糊塗

日本市場學在釐訂價格方面亦方寸大亂，售價訂到他們認爲市場可以接受的最高點，而非儘量將價格壓低，例如 NEC 的 PC—8000 售價爲五百九十五鎊，與 APPLE Ⅱ 差不多，但蘋果却提供額外的特點，其他商業文儀系統，售價是市場上的最高價。日立的日本公司與英國子公司曾在售價上激烈爭吵，英國子公司堅持要減價才能開展市場。

賈拉咸總結日本公司市場學上基本錯誤是日本公司認爲英國微型電腦市場就如日本市場一樣，買家對產品設計及需要並無分別。

爲甚麼一定要外國人採用日本的字符及適應日本製造廠所訂出的工作程序呢？

毫無疑問，日本的微型電腦市場很龐大，但這只不過是本土市場，日本的小型企業與英國公司不同。日本典型的辦公室可能充斥着複製的機器，但應用微型電腦還只是初階，大多數商業事務工作仍然用人手書寫處理。所以我們不用奇怪，到現在爲何日本還未能生產出高效率的打字機。

日本買家的要求

在日本，擁有微型電腦是身份象徵，日本家庭顧客並不滿足於燦爛的色彩，眞正的家庭電腦用家都希望微型電腦有更多特色。如有好的繪圖，而硬件外觀設計優美，有好的工藝綫條。日本市場最大要求是16—BIT 系統——並不是用 8088 作基礎的機器，如 IBM PC，同時要有大量記憶空間及奪目的高解度繪圖能力。日本目前最流行的是 NEC PC—9801，是以 8086爲基礎的機器。但迄今這款電腦在英國市場仍未露面。

日本電腦市場，買家最優先考慮是繪圖能力，其次是處理器的能力，然後才是可以應用的現成軟件數量。電腦可以做一點點程式已令顧客滿足，甚少想到軟碟問題。彩色控制器成爲最重要考慮因素。

日本買家目的只購要一部電腦，因此製造商不用担任支援問題，因爲用家不需要這些服務，但同樣的機器運到美國及歐洲銷售就出現阻滯，在小型商業微型電腦市場上，裁了跟斗，他們仍不

由於採用錯誤的市場策略
日本電腦企業無法打開歐美市場

明白過去兩年所造成的錯誤。

企業架構的影响

日本微型電腦企業的架構亦影响了製造商改進他們的地位，絕少數小型日本公司有衝勁去改革市場，而小型企業是美國電腦業發展的重要動力，日本市場全部由超級企業集團所壟斷，新力、日立、錦囊、川崎差不多掌握了日本工業命脈，這些公司通常有附屬公司在相同的市場上互相競爭（例如，日立及其子公司 NISSEI SANYO ，就在英國銷售同類型產品，彼此競爭）或者是彼此有協議而致令事情更錯綜複雜混淆，（ SANYO 及 NIPPON UNIVAC 在英國推出相同產品，但却交由不同的代理商經銷）。

而日本貿易公司亦出現怪現象，可以代理不同生產商的同類型貨品，形成在海外市場，競爭一片混亂。

最後一個因素，亦是日本才獨有的，製造商只接訂單生產，而自己不參與推銷產品， KYOCERA 就是基中一間，它生產 TANDY 100 及差不多完全雷同 NEC PC—8201 的微型電腦，但却沒有自己的銷售網。

驟眼看來，日本的微型電腦企業似乎很簡單，局外人很難瞭解日本超級企業的管運原來是一場糊塗，內情複雜得令人震驚。

這些大型集團所生產的微型電腦種類十分廣泛，由手提式至八吋軟碟驅動器的枱上電腦一應俱全， NEC 的生產綫上有九款微腦，其中一些品種很類似，實難明何以要重覆生產，不過硬件的細少的分別却可滿足日本市場買家的需求。

唯一成功者

在海外市場上，唯一成功的例子是聲寶牌，MZ—80K ，產品是 2001 PET 的複製品，當他們預算推出 MZ—80 A 及 MZ—80B 之前，就可大幅將 MZ—80 K 的售價減低，刺激銷路。現在當銷路出現疲態時，就立即改用 MZ—700 及 MZ—3500 枱上商業機器爲推銷主要力量。價格在市場上很有競爭力。聲寶牌最新的 PC—5000 ，是手提式微型電腦，非常令人感興趣，而價格約是一千三百鎊。

FUJITSUI 及 TOSHIBA 經已更新檢討他們在英國市場的推銷策略，FUJITSUI，日本最大的電腦公司，最近已與 AMDAHL 及 ICL 達成合作協議，生產微型周邊，首先是 BUBCOM—80 ，磁泡記憶器。微型周邊製造商鉅子， IAN DONALDSON ，驕傲地宣佈已與 FUJITSUI 達成協議，推銷M—8家庭微型電腦。

FUJITSUI 現在正自行推銷 MICRO 16 S，調派日本高級首腦到英國負責成立推銷網。

TOSHIBA ，以前曾出過T200，CP／M機器，最近與 SCAN DATA 簽約（以德州儀器及 ONYX 著名），負責推銷 T 100 及 16—BIT T 300。平價產品將不復再。

NEC 在英國市場將增添三類產品，分別是 TANDY 100，（與 PC—8201 差不多）， APC（8086 機器，附八吋驅動器） PC— 8800，（8 BIT ，替代已有五年歷史的舊式 PC—8000）。

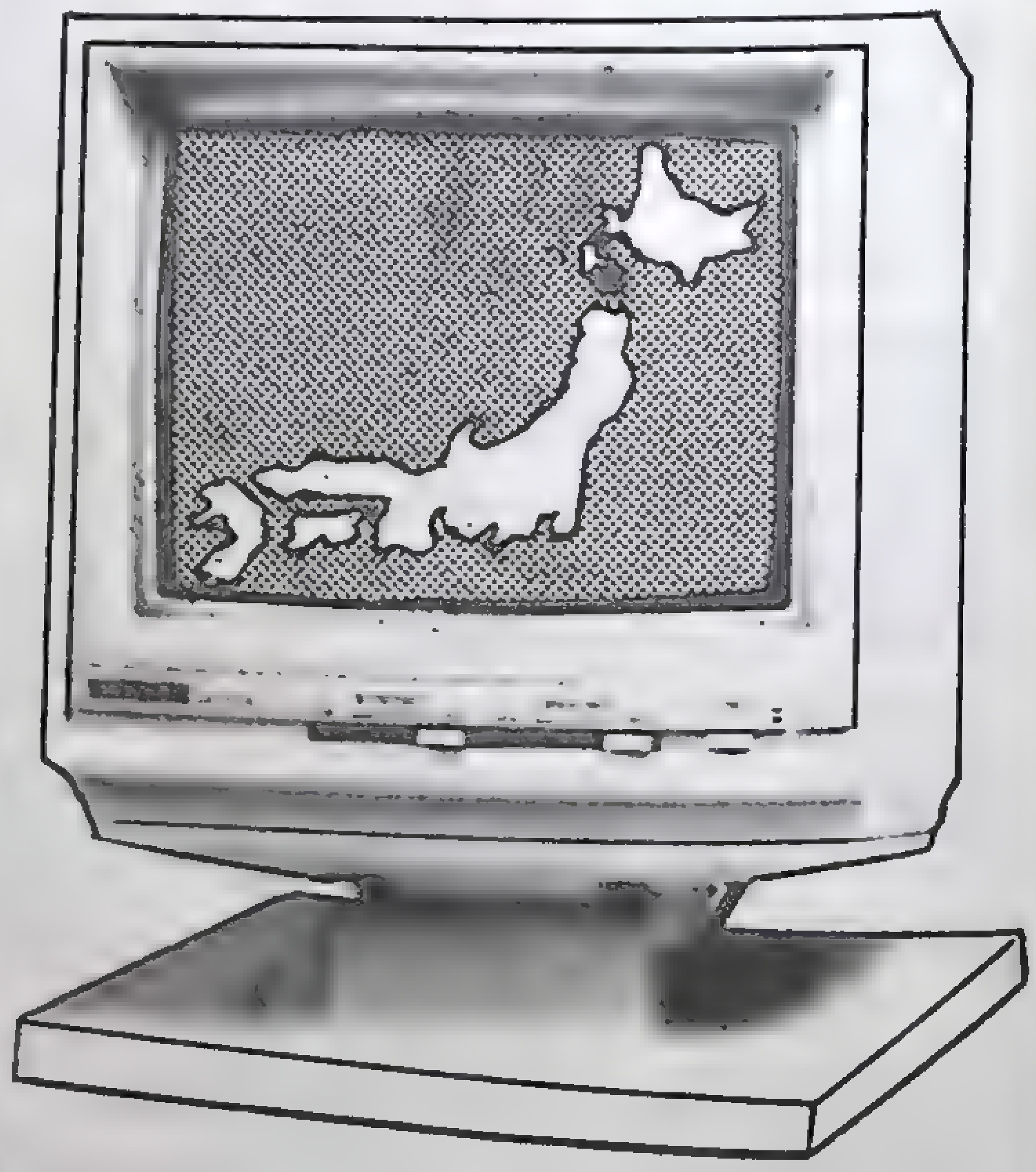
新力將介紹 SMC—70， 及將市場集中推銷 OEM 3 吋半驅動器，E—PSON 推出 HX—20 AND QX—10枱上電腦， SANYO 的四種機器售價很吸引，但很少被人推薦採用， SORD 的系統價格太高， PANA—SONIC期待已久的手提式及彩色家庭機器仍未面世，精工將推出以 8086 爲基礎的8600。

日本公司在微型市場的活躍程度預料不會減弱，但是必須要改善其缺點。

日本本土市場並沒有多大新產品， SORD 的M—5，五色繽紛的機器，推出後却沒有預期流行。明顯地，日本公司在微型電腦行業須要學習的事有很多，至少須確認現成的軟件配合是必需。但這需要美國兩間公司 MICROS—OFT 及 SPECTRAVIDEA ，去提高日本產品的水準， SPECTRAV—IDEA SV—318 及 SV—328 是在英國首先推出的MSX 機器。

到目前爲止，並無跡象顯示，日本的電腦侵略已爲氣候，日本公司熱衷於製造第五代電腦產品，原因不單是希望用聲音科技克服打字問題，更重要是開創微型電腦新市場，而起領導作用。

很多人爭論日本商戰對英國市場所受的打擊，但以個人意見，日本電腦侵略暫仍不足爲患。□



修補DOS

甘貴祥

儘管 DOS 提供了極多的功能給予 BASIC 及滙編語言程式設計員使用，有時用家仍會要永對 DOS 做些變更。因 DOS 經常有新版本推出，且每數個月對 DOS 修改一次，亦是沉重的負担，所以修改 DOS 的要求只在絕對必要的情形下才會接受。大幅度修改 DOS 必須澈底了解每個變更的完全函意及所有的牽連，否則很可能使系統變得不可靠。

一、副磁碟及主磁碟的修補

一般修改 DOS 的程式包含"修補（PATCHING）" DOS 的機器語言碼。當決定了所須要的修改後，DOS 的一些指令會被另外一些修正程式所取代。有三種修改 DOS 的方式可用。

（1）只修補在記憶體中的 DOS，這種修改在 R—EBOOT之後便失效。

（2）執行第一種方式所修補的 DOS 並以它起始設定一磁碟以得到 DOS 永久的修改。就是重新建立（CRE—ATE)一磁碟，將 DOS 複製在它第0，1，2磁軌上。每當 BOOT 新建磁碟，修補過的 DOS版本便載入記憶體，往後在此磁碟上所產生的任何副磁碟也都擁有修補過的D—OS 版本。

（3）直接修補主磁碟。這種方式較麻煩，要利用 Z—AP 程式直接修補主磁碟前三條磁軌中的 DOS 映象，或者利用 MASTER CREATE 將已修改過的 DOS複製寫入新磁碟。下列程式執行此工作：

BLOAD MASTER CREATE

進入監督程式（CALL—151）

將$4C 存入 $80D 位置中（80D：4C）

執行MASTER CREATE（800G）

當 MASTER CREATE 將 DOS 映象載入記憶體後便跳出，然後使用者可以利用監督程式修改 DOS 映象。MASTER CREATE 首先將DOS 的 BOOT2（RWT—S）首先自記憶體 $1200 載入，接着將 DOS 主體自 $ IC 00 載入。

修補完後重入 MASTER CREATE $ 82D 位置（82 DG）。

MASTER CREATE 正確地執行完後，磁碟就完成了修補的工作。

這程序可用在 DOS 3.2，3.2.1及3.3版本上。

二、語言卡的避免重載入

在 DOS 3.3之中作增訂會有個較困擾的地方。BOOT 2 機器碼中有個修補，是將語言卡第一個 BYTE 存入二進位制"零"，以使得不論是否爲開機狀態，每次 BOOT 碟都要重載入 BASIC（INTEGER或APPLESOFT)。開機時這個修補沒有作用，因爲語言卡上第一個 BYTE 並未呈現於 DOS之前供判定爲 BASIC，不論如何此時 BASIC 都須重載入記憶體，但在其後的 BOOT 中，往往記憶體已有完整的 BASIC，而此修補在 BOOT磁碟時仍造成 LANGUAGE NOT AVAILABLE（語言仍未適用）的錯誤信號。DOS 3.3的這個修補原是考慮到可能發生類如 PASCAL 語言的第一 BYTE 恰巧同於B—ASIC 語言的第一 BYTE 之情形，這結果會在 DOS 中造成怪異的現象。若使用者在交換使用 DOS 及其他系統時總是開機再開機，可將這個修補的下列方法拿掉不用：

將$BFD3（48K）3 BYTE 的STA 指令改成三個 NO OP 指令：BFD3：EA EA EA以此修改過的 DOS 起始設定一副磁碟可得新修補 DOS 磁碟，此STA 指令在 32 KDOS 系統中位於 7FD3， 在16K DOS 系統中位於3FD3。

三、DOS緩衝器

有時須要一個安全的區域存放常用的機器碼程式（如：印字機推動器（PRINTER DRIVER)等使其不受 B—ASIC 及 DOS 干擾亦不受 DOS冷啓動（COLDST—ART）影响。短過200 BYTE 的程式可將它存入自 $ 300 開始的位置。若程式大於 200 BYTE，最好將它"塞"入 DOS 及 DOS 緩衝器之間（此程式須爲可重定位（RELO CATABLE）並可在新位置執行)，首先將程式載入到記憶體低位置區再將它複製到低於 $ 9D00（48K）及緩衝器上方，然後更改 $ 9D00內容，使第一個緩衝器的位址指向插入程式的下方（須多空下存檔案名稱及鏈結欄（LINK FIELD）所佔的 38 BYTE 位置），並將程式跳至 3D3（DOS冷啓動）使 DOS 在插入程式的下方另建緩衝器。在下一次 BOOT 之前，DOS 會把插入程式所佔空間忽略掉。因 HIMEN 低於 DOS 緩衝器，BASIC 不會使用那裡的記憶體。

四、BRUN或EXEC命令執行HELLO

當DOS 完全了 BOOT 並進入記憶體後，它執行了在 FILE 案名稱緩衝器中 HELLO FILE 案的 RUN命令（其左爲將 DOS 寫入磁碟的 INIT 命令）。以下方法修補 DOS 可將 RUN 命令改爲 BRUN或 EXEC 。

9E 42：34（改爲 BRUN）

或9E 42：14（改爲 EXEC）

五、實踐例子程式

各位至此應對修改 DOS 的工作，應有些基本概念了，爲此，不妨參考本文末的程式來作一次實踐。

本程式的範疇是將磁碟上特定的 TRACK 及 SEC－TOR 上的資訊經由 RWTS 網路貯入監督程式的 $ 2000 始，在這裡，閣下可以對它們充份視察，修改，而一切的 DOS 及監督程式的命令仍然可用。

因爲我們知道磁碟上資訊的修改，編輯都是以 BYTE 形式來進行，因此有必要利用機械語言作爲溝通橋樑。每次用機械語言編輯完成後，你會回到 BASIC 上去，而按下【RETURN】將停止執行本程式。但若要繼續執行的話，只需按下"E"便可繼續編輯了。

六、使用須知

①本程式所輸入的 TRACK 及 SECTOR 數目須爲十進制。

② SECTOR 內的資訊貯在監督程式位址 $ 2000 至 2OFF

HAPPY EXECUTING。□

```
QUICK ZAPPER

10  REM
100  HOME : GOSUB 1500:IO = 47080

110  PRINT "USE DECIMAL VALUES": PRINT
     : INPUT "TRACK? ";TR: IF TR <
     0 THEN  PRINT : PRINT "ABORT
     ED": END
120  INPUT "SECTOR? ";SE: IF CM =
     2 THEN  RETURN : REM
200  POKE IO + 4,TR: POKE IO + 5,
     SE
210  PRINT : INVERSE : PRINT " SE
     LECT: ";: NORMAL : PRINT " Q
     UIT [0] RERAD [1] WRITE [2]
     ";: GET CM$:CM =  VAL (CM$):
      ON CM = 0 GOTO 800: ON CM =
     2 GOSUB 3000: IF CM <  > 1 AND
     CM <  > 2 THEN 210
220  POKE IO + 12,CM
230 BU = 8192: POKE IO + 8,BU - 2
     56 *  INT (BU / 256): POKE I
     O + 9,BU / 256: REM
240  POKE IO + 3,0: POKE 5,0: REM
      ERR CODE
300  CALL 768: REM
310  IF  PEEK (5) THEN  PRINT : PRINT
     "SORRY, ERROR " PEEK (5)" OC
     CURED": STOP : REM
500  ON CM GOSUB 2000,2010: PRINT
     : IF CM = 2 THEN  END : REM

600  PRINT "SECTOR WAS LOADED AT
     $2000 20FF AND MAY NOW BE EX
     AMINED AND MODIFIED WITH ANY
      MONITOR COMMAND. YOU WILL B
     E RETURNED TOBASIC AFTER EAC
     H EDIT AND MAY WRITE THE SEC
     TOR BACK TO DIAK OR CONTINUE
      EDITING": PRINT
610  GOSUB 1000
620  PRINT : PRINT : PRINT "CONTI
     NUE EDIT OR RETURN TO PROGRA
     M [E/R]";: GET A$
630  ON A$ = "E" GOTO 610: GOTO 2
     10: REM
800  PRINT : END : REM
1000 HEX$ = "": HTAB 1: POKE 51,1
     70: CALL 64874: FOR I = 1 TO
     239:CHR =  PEEK (511 + I): ON
     CHR = 141 GOTO 1510:HEX$ = H
     EX$ +  CHR$ (CHR - 128): NEXT
     : GOTO 1510
1500 HEX$ = "300:20 E3 3 20 B5 B7
      90 5 AD F5 B7 85 5 A9 0 85
     48 60": REM
1510 HEX$ = HEX$ + " N D9C6G": FOR
     I = 1 TO  LEN (HEX$): POKE 5
     11 + I, ASC ( MID$ (HEX$,I))
      + 128: NEXT : POKE 72,0: CALL
     - 144: RETURN : REM
2000  GOSUB 2500: PRINT " READ ":
      RETURN
2010  GOSUB 2500: PRINT " WRITTEN
      TO DISK": RETURN : REM
2500  PRINT : PRINT : PRINT "TRAC
     K "TR", SECTOR "SE;: RETURN

3000  GOSUB 110: PRINT : PRINT : FLASH
     : PRINT "OK TO WRITE TO DISK
      [Y/N] ";: NORMAL : GET A$: IF
     A$ <  > "Y" THEN  POP : HOME
     : GOTO 210
3010  RETURN
```

CP/M 1
入門基礎篇

馮致遠

CP/M系統的闡釋及其應用

CP/M是微處理器的操作系統(CONTROL PROGRAM/MICROPROCESSOR),由廸吉多研究公司(DIGITAL RESEARCH)編滙和發行。CP/M包括了一系列不同功用的程式,使它成爲完美的操作系統;也包括許多不同類別的應用程式,能迎合每一個電腦用戶的需求。CP/M在硬件方面只須一般的微電腦系統,但必須有終端機(螢幕顯示器和鍵盤)、一個或多個磁碟機、和電腦本身;也可加其他附件,如印字機、讀卡器、打卡器等,用家可以自由選擇和組合。

電腦本身和所有軟件都有很密切的關係,由於大部份CP/M的版本是用 8080一滙編程式所寫,電腦本身必須有 8080 微處理器。 INTEL 8085和 ZILOG Z 80 是後期製造的。 8085用的指令和 8080 大致相同,但它的綫路較快; Z 80 具有 8080 指令集以外的數十個指令。所以CP/M能在 8085 和 Z80上運行。在一九八〇年,CP/M的創新版本CP/M—86面世。這版本是專爲8086所寫的,與原來版本大有分別,故目前尙未流行。用 8086 微處理器的最大優點是它能用到 128 KB 記憶體,比以往的 64 KB 多出一倍,能減少許多由磁碟到記憶體的傳遞。相信CP/M—86將在數年內流行。

除了CP/M 2.2 和CP/M—86外,較早期的還有CP/M 1.4 和CP/M 2·0。MP/M1 和MP/M 2是和CP/M非常類似的系統,但MP/M是專爲電腦分時系統(TIME SHARING)所寫,能同時替多個用家工作,而每個都感覺擁有一個CP/M系統。

當CP/M所需的硬體裝備妥當後,試把CP/M系統碟放入磁碟機,然後開動電腦。首先被讀入和運行的程式就是CP/M的根——監督程式(MONITOR),它給予用家和其他程式許多服務,它是由三部份組成的:

一、輸入指令處理程式(CCP)

它在用家和操作系統之間作橋樑。當用家用鍵盤打入指令後,CCP便到磁碟上的指令檔案錄中找這檔案;把它讀入記憶體中,然後運作。當指令運作完畢,它會再被讀入和工作。

二、基本磁碟操作程式(BDOS)

包括了三十多個副程式,是應用程式和系統之間的橋樑。它主要是磁碟和檔案的操作副程式,也有印字機、鍵盤和螢幕的輸入輸出副程式等。可說是CP/M提供所有用家和程式的功能。但不同版本的功能在總數上和功能上都有分別。

三、基本輸入及輸出程式(BIOS)

由於不同型號電腦的硬件設計和原理都有很大分別,BDOS 的副程式必須在 BIOS 的支持下才能控制電腦硬件和資料的轉遞;是系統和硬體的橋樑。 BIOS 是依照不同硬體設計而編寫的,不同的電腦自然需要不同的 BIOS 了。

其實CP/M監督程式的最大功用是提供橋樑給用家和電腦,使應用程式能簡易地控制和運用電腦和附件。

明顯地,單有這個監督程式是不足以成爲一個操作系統。CP/M還包括了一些檔案操作程式,使用家能簡易地處理磁碟上的檔案:

一、DIR

用來列出所有或某些儲在指定磁碟機上的檔案名稱和類別。

二、ERA

可把一個或多個在某磁碟上的檔案刪去。這些必須是可讀可寫的檔案;唯讀檔案是不會被刪掉的。

三、REW

可更改某檔案的名稱及類別。CP/M檔案的名稱以最多八個字母代表,如(TESTFILE) 而它的類別以最多三個字母代表。如 COM,TXT 等。無論以大楷或小楷輸入, CCP 必先把全部轉爲大楷才錄在碟上。

四、SAVE

用來把記憶體以1/4K(256 個字節)作單位,儲在磁碟上。它是由100H 開始抄錄的,因爲這位置是程式工作區的起點。這程式必定是 CCP 的一部份,若它需要先由磁碟讀入,記憶體的值將會被改變了。

五、TYPE

能把磁碟上的文字檔案列在螢幕上,程式的運作能以 CTRL—S 來暫停,使讀者有充份時間閱讀檔案。只要按下 RETURN鍵,程式的運作便立即終止。若用家先按下 CTRL—P 鍵,檔案將同時在印字機上列出。

六、USER

CP/M可在磁碟上劃分出最多十六個邏輯區(SECTOR)(0—15)。這設備對較小容量磁碟機用家可能沒多大用途,但對大容量磁碟儲存器用家便實際得多了。因爲大量不同的檔案儲在同一隻磁上是不易處理的,如果把它們分門別類;放在不同的邏輯區中集中處理,便能較有條理和方便了。利用這指令,用家便能從一個邏輯區跳到另一個邏輯區工作。每次系統的啓動,大家都是在零邏輯區的。一般小型磁碟機用家都是長期使用這邏輯區。而以不用的磁碟來把檔案分類。

由於以上六個指令都是常用的;所佔記憶體不多,所以都包括在 CCP 裡,毋須由磁碟讀入才工作,提供用家許多方便的同時也替他們節省時間。

至於其他需要由磁碟讀入的指令,五花百門,沒有限制

CP/M 專欄

，也可由用家自己編寫。它們就是磁碟上的 COM 類的檔案，稱爲指令檔案，CP／M也提供了一些非常實用的指令檔案，是必須的工具程式，例如：

一、ED・COM

是文字檔案的編輯程式，可用作程式本文的編滙和修改。

二、PIP・COM

用來協助用家的檔案轉送工作，包括磁碟之間和磁碟與其他裝備之間的轉遞工作。用家可選擇全部或部份的轉遞也可把原來的檔案作部份的修改。

三、STAT・COM

可統計磁碟及磁碟上檔案的詳細資料，如檔案所佔空間、檔案是可以讀寫、磁碟餘下的空間等；也可更改檔案及磁碟的標記（例如：唯讀標記、列出標記）。除此以外，它可列出或改變裝備的邏輯名稱（例如：印字機可稱爲 LST：或 LPT：）；更可指示現在的邏輯區。

用家可自行創作工作程式，因爲CP／M還提供許多不同編滙語言的編譯器，把高級語言所寫的程式翻譯成機械語言程式。

還有許多由軟件公司發展的應用程式，總數達一千五百種以上；無論工商界、程式編滙員、工程師 、電腦愛好者，甚至學生也不難找到理想的軟件。

筆者在以後會基於 APPLE Ⅱ 微電腦的 CP／M 操作系統，與各位讀者作深入淺出的討論，並加以程式的例子，使讀者對 CP／M有更深入的認識及運用，下期再見。■

AP II 微電腦之進化史

（上接第23 頁）

（2）MONITOR新加了三個指令，即“STEP”，“BREAK”及“FUNCTION”

（3）BASIC 也加了新的命令，即“TEST”，“AUTO”及“P—ATN”。

（4）音樂表達方面可以有8度音，39個音階爲了配合這些新的功能，它的鍵盤設計也就改良成：

（A）63個功能鍵。

（B）30個指令鍵。

（C）15個獨立數字鍵。

（D）由鍵盤直接描劃造型表。

由於第六代推出至今的時間還是很短，它的工作特性與及其他介面卡之相容性，則還有一定時間之觀察及實驗才可以知其全貌矣。

(7) 展望將來

市場上現正流行分體式的製作，鍵盤與主機爲兩個個體，其間的聯絡則倚靠一條引綫相連接，鍵盤的設計除了有功能鍵，用家設計鍵及獨立數字鍵外，整個鍵盤還可以任意擺放，及以任何角度的調整，這比較整體結構化的古老設計來得方便及靈活得多。主機方面，空間大了，有更多地方可以裝置一至四部的磁碟機，介面卡、叭喇及電源供應器等，抽氣扇也可以直接裝在主機內，令到熱空氣更能有效地抽出機外。將來（第七代？）的 AP Ⅱ會不會是分體式的設計呢？或是有更多更新的特性附加呢？我們且拭目以待吧！■

一九八三年 十大暢銷電腦製造商 IBM獨佔鰲頭

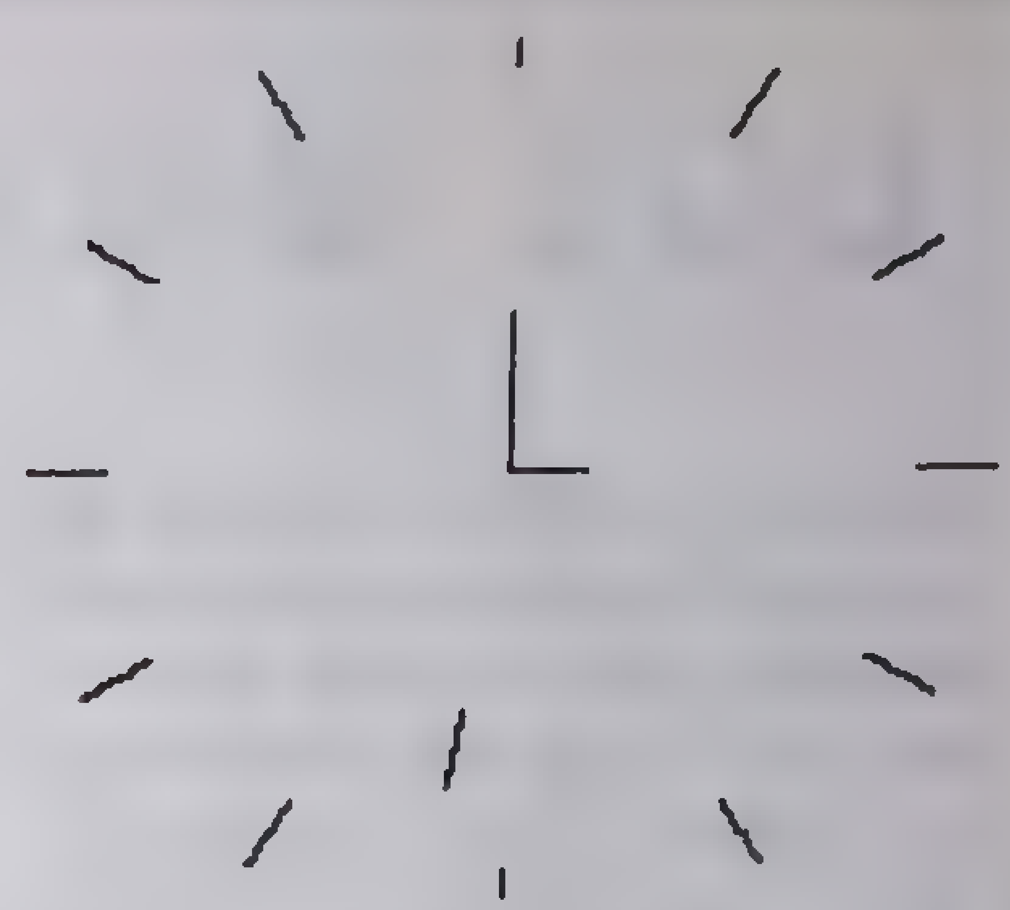
英國一家市場統計調查公司最近發表了一項調查公佈，頭十家最暢銷私人電腦製造商名單及其營業額。他們的名次如下：

IBM	$1400
APPLE COMPUTER	1100
RADIO SHACK	1100
COMMODE	800
HEWLETT—PACKARD	550
TEXAS INSTRUMENTS	450
ATARI	350
DIGITAL EQUIPMENT CORP	200
VICTOR	150
TELEVIDEO	100

（百萬美元計）

這一次，所有這十家製造商營業額均能突破一億美元。在一九八二年，只有三家公司有此營業數字。估計全年硬件及軟件的銷售額超過十億美元。値得注意的入選十大的製造商中，有三家是新加入製造私人電腦行業的。他們包括：DEC、VICTOR 及 TELEVIDEO 。■

黑房計時器



APPLE的 REAL TIME CARD 在本港並不太流行，要花上二百元來購買不太值得，因此筆者寫了這一個簡單程式來模擬時鐘效果，它在短期間內的誤差倒還可以容忍，這樣便符合了本人用作黑房中的計時器之用，作為曝光計時等倒還算得過去，若各位讀者有更佳的應用構思，歡迎來信賜教。

```
]LIST

500  REM  CLOCK FACE
1000  REM
1005  REM  VARIABLES
1010  REM
1015  REM  H,MN,S...START TIME
1020  REM  XI,YI....SHAPE 1 DRAW
     POINT
1025  REM  RH,RM,RS,SHAPE ROTATIO
     NS
1030  REM  LC.......SHAPE TABLE L
     OCATIONS
1035  REM  BT.......SHAPE TABLE B
     ITS
1040  REM  INC......INCREMENT TO
     TAL
1045  REM  I........INCREMENT COU
     NT
1050  REM  B........FIRST RUN LAB
     EL
1500  REM
1505  REM  SET SCREEN POSITION
1510  REM
1515 X1 = 140
1520 Y1 = 96
1525 X2 = X1
1530 Y2 = Y1
1535 X3 = X1
1540 Y3 = Y1
1545 INC = 529
2000  REM
2005  REM  INITIALIZE CLOCK
2010  REM
2015  TEXT : HOME
2020  INVERSE : PRINT "CLOCK": NORMAL

2025  PRINT : PRINT "START TIME"
2030  INPUT "HOUR......:";H
2035  INPUT "MINUTES...:";MN
2040  INPUT "SECONDS...:";S
2500  REM
2505  REM  LOAD SHAPE TABLES
2510  REM
2515  POKE 232,0: POKE 233,3
2520  FOR LC = 768 TO 792
2525  READ BT
2530  POKE LC,BT
2535  NEXT LC
2540  DATA   4,0,10,0,12,0,14,0
2541  DATA   18,0
2545  DATA   4,0,36,0,8,24,32,0,8

2550  DATA   24,8,24,32,0,0
2560  HGR2
2562  SCALE= 17
2565  POKE 16302,0
2570  FOR I = 0 TO 11
2573  READ R
2580  ROT= R
2590  XDRAW 4 AT X1,Y1
2600  NEXT I
2620  DATA  0,6,11,16,21,26,32,38
    ,43
2630  DATA  48,53,58
3000  REM
3005  REM  MAIN PROGRAM
3010  REM
3015  REM
3500  REM  *** DISPLAY CLOCK
3515 B = 1: REM  FIRST RUN LABEL
3520  GOSUB 4000: REM  DRAW HANDS

3525  GOSUB 6000: REM  COUNTER
3530 S = S + 1
3535  IF S = 60 THEN :MN = MN + 1
    :S = 0
3540  IF MN = 60 THEN :H = H + 1:
    MN = 0
3545  IF H = 13 THEN :H = 0
3550  GOSUB 5505
3555  GOTO 3520
4000  REM  DISPLAY HANDS
4005  IF B = 1 THEN : GOTO 5000: REM
     SKIPS UNDRAW ON FIRST RUN
4010  IF S <  > 0 THEN : RETURN
4500  REM  UNDRAW HANDS
4505  ROT= RH
4510  XDRAW 1 AT X1,Y1
4515  ROT= RM
4520  XDRAW 2 AT X2,Y2
5000  REM  DRAW HANDS
5005 RM = MN * 64 / 60
5010  ROT= RM
5015  DRAW 2 AT X2,Y2
5020 RH = H * 64 / 12
5025  ROT= RH
5030  DRAW 1 AT X1,Y1
5035  RETURN
5500  REM  SECONDS DISPLAY
5505  IF B = 1 THEN :B = 0: GOTO
    5515
5510  ROT= RT: XDRAW 3 AT X3,Y3
5515 RT =  INT (S * 64 / 60)
5520  ROT= RT: DRAW 3 AT X3,Y3
5525  RETURN
6000  REM  COUNTER
6005  FOR I = 1 TO INC
6010  NEXT I
6015  RETURN
```

DOS 操作指令簡化器

傳統的DOS 操作命令十分麻煩，須按入很多字符才能執行，爲此我編寫這個程式來簡化使用過程，利用 BASIC 達到目的，以便用 APPEND命令加在任何 BASIC 程式上，或單獨執行，甚至永久改變 DOS，其基本原理爲 POKE 入這些新設定的命令於記憶體中，這裡重新設定的命令有十一個，其它現用的命令則仍維持不變。

現將一些新 DOS 命令更改爲方便使用的字樣，例如：

LOAD	改爲	LO
SAVE	改爲	S
DELETE	改爲	DE
LOCK	改爲	LK
UNLOCK	改爲	UL
RENAME	改爲	RN
CATALOG	改爲	X
IN#	改爲	#
BSAVE	改爲	BS
BLOAD	改爲	BLO
VERIFY	改爲	V

(A) 本程式可以作爲子程式加在任何 BASIC程式之後，只須 RUN 執行它便可以得到所需的控制命令。

(B) 由於本程式爲方便使用，因此以 TEXT FILE 貯存，各位可直接用 EXEC COMMAND CHANGER 來執行之。

(C) 永久性的保留方式：由於 DOS 是貯存於記憶體內，因此在電源關上後便會失去，若經常要採用上述 A、B 兩種方法是十分麻煩的，爲此各位可以在執行本程式後，取一隻新碟或其中資訊不重要者以 INIT 命令初始化，這樣便會將 DOS 命令存入第 0，1，2 TRACK 上，只要 BOOT 此磁碟時便可重新取回這些 DOS 命令了。

```
]LIST

30000 M = ( PEEK (978) - ( PEEK (
     978) > 127) * 256) * 256 + 2
     947:E = M + 240:AE = 174 - (
     157 -  PEEK (978))
30010  REM  DOS COMMANDS
30011  POKE M + 1,73: POKE M + 2,
     78: POKE M + 3,73: POKE M +
     4,212: POKE M + 5,76: POKE M
      + 6,207: POKE M + 7,211: POKE
     M + 8,82:
30012  POKE M + 9,85: POKE M + 10
     ,206: POKE M + 11,67: POKE M
      + 12,72: POKE M + 13,65: POKE
     M + 14,73: POKE M + 15,206: POKE
     M + 16,68:
30013  POKE M + 17,197: POKE M +
     18,76: POKE M + 19,203: POKE
     M + 20,85: POKE M + 21,204: POKE
     M + 22,67: POKE M + 23,76: POKE
     M + 24,79:
30014  POKE M + 25,83: POKE M + 2
     6,197: POKE M + 27,82: POKE
     M + 28,69: POKE M + 29,65: POKE
     M + 30,196: POKE M + 31,69: POKE
     M + 32,88:
30015  POKE M + 33,69: POKE M + 3
     4,195: POKE M + 35,87: POKE
     M + 36,82: POKE M + 37,73: POKE
     M + 38,84: POKE M + 39,197: POKE
     M + 40,80:
30016  POKE M + 41,79: POKE M + 4
     2,83: POKE M + 43,73: POKE M
      + 44,84: POKE M + 45,73: POKE
     M + 46,79: POKE M + 47,206: POKE
     M + 48,79:
30017  POKE M + 49,80: POKE M + 5
     0,69: POKE M + 51,206: POKE
     M + 52,65: POKE M + 53,80: POKE
     M + 54,80: POKE M + 55,69: POKE
     M + 56,78:
30018  POKE M + 57,196: POKE M +
     58,82: POKE M + 59,206: POKE
     M + 60,216: POKE M + 61,77: POKE
     M + 62,79: POKE M + 63,206: POKE
     M + 64,78:
30019  POKE M + 65,79: POKE M + 6
     6,77: POKE M + 67,79: POKE M
      + 68,206: POKE M + 69,80: POKE
     M + 70,82: POKE M + 71,163: POKE
     M + 72,163:
30020  POKE M + 73,77: POKE M + 7
     4,65: POKE M + 75,88: POKE M
      + 76,70: POKE M + 77,73: POKE
     M + 78,76: POKE M + 79,69: POKE
     M + 80,211:
30021  POKE M + 81,70: POKE M + 8
     2,208: POKE M + 83,73: POKE
     M + 84,78: POKE M + 85,212: POKE
     M + 86,66: POKE M + 87,211: POKE
     M + 88,66:
30022  POKE M + 89,76: POKE M + 9
     0,207: POKE M + 91,66: POKE
     M + 92,82: POKE M + 93,85: POKE
     M + 94,206: POKE M + 95,214:
      POKE M + 96,171:
30023  POKE M + 97,171: POKE M +
     98,171: POKE M + 99,171: POKE
     M + 100,171: POKE M + 101,17
     1: POKE M + 102,171: POKE M +
     103,171: POKE M + 104,171:
30024  POKE M + 105,171: POKE M +
     106,171: POKE M + 107,171: POKE
     M + 108,171: POKE M + 109,17
     1: POKE M + 110,171: POKE M +
     111,171: POKE M + 112,171:
30025  POKE M + 113,171: POKE M +
     114,171: POKE M + 115,171: POKE
     M + 116,171: POKE M + 117,17
     1: POKE M + 118,171: POKE M +
     119,171: POKE M + 120,171:
30026  POKE M + 121,171: POKE M +
     122,171: POKE M + 123,171: POKE
     M + 124,171: POKE M + 125,17
     1: POKE M + 126,171: POKE M +
     127,171: POKE M + 128,171:
30027  POKE M + 129,171: POKE M +
     130,171: POKE M + 131,171: POKE
     M + 132,171:
30900  RETURN
```

■

找回APPLE II PLUS 失落了的STEP和TRACE

我們的 APPLE Ⅱ PLUS是一部非常卓越的機器，事實上它的編輯能力比標準的 APPLE Ⅱ好，例如 CURSOR 控制，有停止列印功能 APPLE SOFT 的效能更宏大，是一種很好的語言，因此與 INTEGER BASIC 相比，我較爲偏愛 APPLE SOFT BASIC，不過由於 ROM 的空間有限，加入了新的功能必須犧牲一些原有的功能，結果，以往 APPLE Ⅱ 機器語言的 STEP／TRACE 功能因爲記憶空間不足而摒棄。

我不願因此而損失了 STEP 和 TRACE。它允許你逐步的執行機械語言的每一指令，並顯示 6502 所有暫存器的値，因此你可以發現存在程式中的錯誤，或只看程式如何工作， STEP 每次執行一個指令，而 TRACE 是連續的，除非碰到 BRK 指令才停止。

使用方法

STEP／TRACE 能在監督系統加上 STEP 和 TRACE 的功能，它的動作和監督系統使用法和原來的 APPLE 監督系統一樣，打入一個十六進位的位址，並接著一個或多個 "S" 便會從那位址開始一步步的執行，打入一個十六進位的位址並指著一個 "T" 就會從此位址開始追踪程式。

S和T比原來的 APPLE Ⅱ 效果更好，因爲要停止的話只要按任何鍵即可（例如 RETURN），要繼續時，只須打入 "T" 便會追踪程式，而且從剛停止的位址繼續執行，或者你也可以打入 "S" 一步步的測試，在S和T系統中我們以反亮的 "X" 爲提示符號，以區別正常 APPLE 監督系統。S和T除了 STEP 和 TRACE 外，也包括了所有的監督系統命令。事實上它亦利用很多監督系統的功能來完成此工作。

程式的結構

程式存於位址 $300至$3E9，和 DOS 的位址指標發生重叠現象，因此本程式移到其他位址即可避免此問題，不過在 69，75，83，91，120，168以及169行，亦即 $328，$337，$349，$35A，$398，$3E7 等位址上的跳越位址須作適當調整。在這裡我們已解決了這一問題。總括而言 STEP／TRACE 在你的機械語言裡是很有用的。

使用本程式的方法是先將 STEP／TRACE 載入到記憶體中，然後 CALL 768 執行之，或直接從磁碟打入 "BRUN"即可，除了 STEP 和 TRACE 外，還可以使用任何監督系統的命令，若要跳開此程式，只須按 "RESET" 即可。□

STEP-TRACE, STEP-TRACE 2 are the same program but start at different address where STEP-TRACE 2 is designated to use together with TRACER.

STEP-TRACE

```
*300LLLLL

0300-   D8          CLD
0301-   20 3A FF    JSR   $FF3A
0304-   A9 2A       LDA   #$2A
0306-   85 33       STA   $33
0308-   20 67 FD    JSR   $FD67
030B-   20 C7 FF    JSR   $FFC7
030E-   20 A7 FF    JSR   $FFA7
0311-   84 34       STY   $34
0313-   C9 EC       CMP   #$EC
0315-   F0 0B       BEQ   $0322
0317-   C9 ED       CMP   #$ED
0319-   D0 0F       BNE   $032A
031B-   AD 00 C0    LDA   $C000
031E-   30 24       BMI   $0344
0320-   C6 34       DEC   $34
0322-   20 C7 FF    JSR   $FFC7
```

```
0325-   20 49 03   JSR   $0349
0328-   10 1A      BPL   $0344
032A-   C9 C6      CMP   #$C6
032C-   D0 09      BNE   $0337
032E-   20 C5 FF   JSR   $FFC5
0331-   20 00 FE   JSR   $FE00
0334-   4C 04 03   JMP   $0304
0337-   A0 17      LDY   #$17
0339-   88         DEY
033A-   30 C4      BMI   $0300
033C-   D9 CC FF   CMP   $FFCC,Y
033F-   D0 F8      BNE   $0339
0341-   20 BE FF   JSR   $FFBE
0344-   A4 34      LDY   $34
0346-   4C 0E 03   JMP   $030E
0349-   20 75 FE   JSR   $FE75
034C-   20 D0 F8   JSR   $F8D0
034F-   68         PLA
0350-   85 2C      STA   $2C
0352-   68         PLA
0353-   85 2D      STA   $2D
0355-   A2 08      LDX   #$08
0357-   BD E1 03   LDA   $03E1,X
035A-   95 3C      STA   $3C,X
035C-   CA         DEX
035D-   D0 F8      BNE   $0357
035F-   A1 3A      LDA   ($3A,X)
0361-   F0 2C      BEQ   $038F
0363-   A4 2F      LDY   $2F
0365-   C9 20      CMP   #$20
0367-   F0 43      BEQ   $03AC
0369-   C9 60      CMP   #$60
036B-   F0 2F      BEQ   $039C
036D-   C9 4C      CMP   #$4C
036F-   F0 46      BEQ   $03B7
0371-   C9 6C      CMP   #$6C
0373-   F0 43      BEQ   $03B8
0375-   C9 40      CMP   #$40
0377-   F0 1F      BEQ   $0398
0379-   29 1F      AND   #$1F
037B-   49 14      EOR   #$14
037D-   C9 04      CMP   #$04
037F-   F0 02      BEQ   $0383
0381-   B1 3A      LDA   ($3A),Y
0383-   99 3C 00   STA   $003C,Y
0386-   88         DEY
0387-   10 F8      BPL   $0381
0389-   20 3F FF   JSR   $FF3F
038C-   4C 3C 00   JMP   $003C
038F-   20 82 F8   JSR   $F882
0392-   20 DA FA   JSR   $FADA
0395-   4C 00 03   JMP   $0300
0398-   18         CLC
0399-   68         PLA
039A-   85 48      STA   $48
039C-   68         PLA
039D-   85 3A      STA   $3A
039F-   68         PLA
03A0-   85 3B      STA   $3B
03A2-   A5 2F      LDA   $2F
03A4-   20 56 F9   JSR   $F956
03A7-   84 3B      STY   $3B
03A9-   18         CLC
03AA-   90 14      BCC   $03C0
03AC-   18         CLC
03AD-   20 54 F9   JSR   $F954
03B0-   AA         TAX
03B1-   98         TYA
03B2-   48         PHA
03B3-   8A         TXA
03B4-   48         PHA
03B5-   A0 02      LDY   #$02
03B7-   18         CLC
03B8-   B1 3A      LDA   ($3A),Y
03BA-   AA         TAX
03BB-   88         DEY
03BC-   B1 3A      LDA   ($3A),Y
03BE-   86 3B      STX   $3B
03C0-   85 3A      STA   $3A
03C2-   B0 F3      BCS   $03B7
03C4-   A5 2D      LDA   $2D
03C6-   48         PHA
03C7-   A5 2C      LDA   $2C
03C9-   48         PHA
*LL

03CA-   4C D7 FA   JMP   $FAD7
03CD-   18         CLC
03CE-   A0 01      LDY   #$01
03D0-   B1 3A      LDA   ($3A),Y
03D2-   20 56 F9   JSR   $F956
03D5-   85 3A      STA   $3A
03D7-   98         TYA
03D8-   38         SEC
03D9-   B0 C5      BCS   $03A0
03DB-   20 4A FF   JSR   $FF4A
03DE-   38         SEC
03DF-   B0 C1      BCS   $03A2
03E1-   EA         NOP
03E2-   EA         NOP
03E3-   EA         NOP
03E4-   4C DB 03   JMP   $03DB
03E7-   4C CD 03   JMP   $03CD
03EA-   4C 51 A8   JMP   $A851
```

STEP-TRACE 2

```
*800LLLLL

0800-   D8          CLD
0801-   20 3A FF    JSR   $FF3A
0804-   A9 2A       LDA   #$2A
0806-   85 33       STA   $33
0808-   20 67 FD    JSR   $FD67
080B-   20 C7 FF    JSR   $FFC7
080E-   20 A7 FF    JSR   $FFA7
0811-   84 34       STY   $34
0813-   C9 EC       CMP   #$EC
0815-   F0 0B       BEQ   $0822
0817-   C9 ED       CMP   #$ED
0819-   D0 0F       BNE   $082A
081B-   AD 00 C0    LDA   $C000
081E-   EA          NOP
081F-   EA          NOP
0820-   C6 34       DEC   $34
0822-   20 C7 FF    JSR   $FFC7
0825-   20 49 08    JSR   $0849
0828-   10 1A       BPL   $0844
082A-   C9 C6       CMP   #$C6
082C-   D0 09       BNE   $0837
082E-   20 C5 FF    JSR   $FFC5
0831-   20 00 FE    JSR   $FE00
0834-   4C 04 08    JMP   $0804
0837-   A0 17       LDY   #$17
0839-   88          DEY
083A-   30 C4       BMI   $0800
083C-   D9 CC FF    CMP   $FFCC,Y
083F-   D0 F8       BNE   $0839
0841-   20 BE FF    JSR   $FFBE
0844-   A4 34       LDY   $34
0846-   4C 0E 08    JMP   $080E
0849-   20 75 FE    JSR   $FE75
084C-   20 D0 F8    JSR   $F8D0
084F-   68          PLA
0850-   85 2C       STA   $2C
0852-   68          PLA
0853-   85 2D       STA   $2D
0855-   A2 08       LDX   #$08
0857-   BD E1 08    LDA   $08E1,X
085A-   95 3C       STA   $3C,X
085C-   CA          DEX
085D-   D0 F8       BNE   $0857
085F-   A1 3A       LDA   ($3A,X)
0861-   F0 2C       BEQ   $088F
0863-   A4 2F       LDY   $2F
0865-   C9 20       CMP   #$20
0867-   F0 43       BEQ   $08AC
0869-   C9 60       CMP   #$60
086B-   F0 2F       BEQ   $089C
086D-   C9 4C       CMP   #$4C
086F-   F0 46       BEQ   $08B7
0871-   C9 6C       CMP   #$6C
0873-   F0 43       BEQ   $08B8
0875-   C9 40       CMP   #$40
0877-   F0 1F       BEQ   $0898
0879-   29 1F       AND   #$1F
087B-   49 14       EOR   #$14
087D-   C9 04       CMP   #$04
087F-   F0 02       BEQ   $0883
0881-   B1 3A       LDA   ($3A),Y
0883-   99 3C 00    STA   $003C,Y
0886-   88          DEY
0887-   10 F8       BPL   $0881
0889-   20 3F FF    JSR   $FF3F
088C-   4C 3C 00    JMP   $003C
088F-   20 82 F8    JSR   $F882
0892-   20 DA FA    JSR   $FADA
0895-   4C 00 08    JMP   $0800
0898-   18          CLC
0899-   68          PLA
089A-   85 48       STA   $48
089C-   68          PLA
089D-   85 3A       STA   $3A
089F-   68          PLA
08A0-   85 3B       STA   $3B
08A2-   A5 2F       LDA   $2F
08A4-   20 56 F9    JSR   $F956
08A7-   84 3B       STY   $3B
08A9-   18          CLC
08AA-   90 14       BCC   $08C0
08AC-   18          CLC
08AD-   20 54 F9    JSR   $F954
08B0-   AA          TAX
08B1-   98          TYA
08B2-   48          PHA
08B3-   8A          TXA
08B4-   48          PHA
08B5-   A0 02       LDY   #$02
08B7-   18          CLC
08B8-   B1 3A       LDA   ($3A),Y
08BA-   AA          TAX
08BB-   88          DEY
08BC-   B1 3A       LDA   ($3A),Y
08BE-   86 3B       STX   $3B
08C0-   85 3A       STA   $3A
08C2-   B0 F3       BCS   $08B7
08C4-   A5 2D       LDA   $2D
08C6-   48          PHA
08C7-   A5 2C       LDA   $2C
*L

08C9-   48          PHA
08CA-   4C D7 FA    JMP   $FAD7
08CD-   18          CLC
08CE-   A0 01       LDY   #$01
08D0-   B1 3A       LDA   ($3A),Y
08D2-   20 56 F9    JSR   $F956
08D5-   85 3A       STA   $3A
08D7-   98          TYA
08D8-   38          SEC
08D9-   B0 C5       BCS   $08A0
08DB-   20 4A FF    JSR   $FF4A
08DE-   38          SEC
08DF-   B0 C1       BCS   $08A2
08E1-   EA          NOP
08E2-   EA          NOP
08E3-   EA          NOP
08E4-   4C DB 08    JMP   $08DB
08E7-   4C CD 08    JMP   $08CD
08EA-   EA          NOP
```

TRACER控制常式

APPLE的 STEP／TRACE 是非常便利的工具，但是如果你將它和此 TRACER 程式連起來使用，將會發現更具威力。

在 APPLE Ⅱ ROM 中的監督系統中包容有許多有用的常式，包括記憶體的詢問與修改、鍵盤輸入、CRT 顯示輸出及卡式帶 I／O。除此之外，APPLE 並且還爲滙編語言程式提供一些常式。一次處理的滙編語言滙編程式（ONE PASS ASSEMBLER）以及解滙編語言程式（DISASSEMBLE）在寫或閱讀機器碼時非常有用。S－TEP／TRACE 的功能允許你控制程式的執行。

STEP 常式一次執行一個指令並顯示它的位址，包括所翻譯的碼，暫存器A、X、Y、P的值，以及堆疊指標。你可以修改任何暫存器的值，並且可以從下一個或任意一個指令繼續執行。

很不幸的，將這些值通通打在螢幕上只能顯示最近的11個指令的狀況。但我們若犧牲一些其他資料而多印一些PC的經歷值，將會更有用。

此 TRACER 程式在使用時要配合 APPLE 的 S－TEP／TRACE 常式，加強他們的功能。基本上這程式的作用主要在控制螢幕的輸出。此常式將印出最近執行的 160 個指令的PC值。當最後一個指令的所有資料印出後（即包括暫存器的値等），你可以繼續執行或暫時停止來修改暫存器或記憶體的值。 TRACER 也有可能碰到中斷碼（00），碰到時會發出二聲BB，然後就等使用者的動作，中斷點之前所執行的最後一個指令也會印出。

注意：TRACER 延遲一個指令顯示。如果經由 RE－SET 進入監督系統，而由於下一個指令已經執行，所以要保持現今暫存器的值很困難，你可以利用監督系統命令C－ONTROL－E查看值對不對。

程式從＄300 開始，計有 190個位元組，它沒有用到第0頁記憶體的其他位置。

TRACER將 STEP／TRACE所產生的字符串處理一下，然後控制那些資訊顯示在螢光幕上。 TRACER將尋找某一特定字符或順序以便可以知道何時才是一個指令的結束。

APPLE 所用的2行顯示格式增加了一點雜複性。執行完一個指令之後，以一 RETURN （$8D）爲開頭，先印出十六進位的位址，此位址即剛剛執行的指令的所在位址，再印出其他的字串。當此行印完後，又接著一個 RE－TURN 然後再印第二行。

TRACER 找到 RETURN 碼之後，經由 COUT 傳送口將第一行的資料移到 TRACER。剩餘的資料則存在緩衝器（ BUFFER） 中，而第二行的資料是一個 RET－URN 碼緊接着一個空白碼（＄A0）來區別。等到下一個RETURN碼出現時，便利用監督系統的 COUT 常式（＄FDF0）將緩衝器中的十六進位的位址（加一個空白）顯示在螢幕上。此常式將會以每劃8個位址共 20 列的格式，顯示160 個位址。

因爲緩衝器碰巧是螢幕記憶體的一部份，所以它也會顯示出來。經由移動自動捲頁視窗（ SCROLLING WI－NDOW） 的底界來保護緩衝器部份。

CONTROL Y 的功能是利用位址 $3F8 的 JMP 指令啓動 TRACER。它將清除螢幕，設立自動捲頁視窗，並且設立在＄36與37的COUT 傳送口，以便將資料傳到TRACER。

TRACER 的用法很簡單：

一、從 $300位址開始載入 TRACER。（不要忘記 CO－NTROL－Y在 3FB 上的 JMP：4C 00 03）。

二、從監督系統執行程式只要打：Yc XXXXT，其中 Yc就是 CONTROL－Y，XXXX 就是開始除錯的位址。螢幕上將會很乾淨， TRACER 已經由COUT 連結好了，並且從指令位址開始追踪。

三、開始時，TRACER只追踪一個指令，並且在螢幕的底端顯示 STEP／TRACE 的資料後會暫停，若要繼續追踪只要按空白鍵即可。剛剛被執行的指令位址也會出現在螢幕最上面。

四、若按 RETURN 鍵，則將會執行一整頁的指令，按下任何鍵將再取得控制權。

五、當然，按下 RESET 鍵將會回到監督系統，但仍可處理暫存器和記憶體的值。再打 YcT 便可再進入 TR－ACER。

TRACER程式列表

```
*300L

0300-   20 3C FB    JSR   $FB3C
0303-   20 58 FC    JSR   $FC58
0306-   A9 15       LDA   #$15
0308-   85 23       STA   $23
030A-   A9 1C       LDA   #$1C
030C-   85 36       STA   $36
030E-   A9 03       LDA   #$03
0310-   85 37       STA   $37
0312-   A9 1F       LDA   #$1F
0314-   85 24       STA   $24
0316-   A9 02       LDA   #$02
```

TRACER

```
0318-    8D BB 03    STA    $03BB
031B-    60          RTS
031C-    8D B7 03    STA    $03B7
031F-    8C B8 03    STY    $03B8
0322-    2C BA 03    BIT    $03BA
0325-    30 1C       BMI    $0343
0327-    C9 8D       CMP    #$8D
0329-    F0 0C       BEQ    $0337
032B-    AC B9 03    LDY    $03B9
*L

032E-    99 50 07    STA    $0750,Y
0331-    C8          INY
0332-    8C B9 03    STY    $03B9
0335-    D0 05       BNE    $033C
0337-    A0 80       LDY    #$80
0339-    8C BA 03    STY    $03BA
033C-    AD B7 03    LDA    $03B7
033F-    AC B8 03    LDY    $03B8
0342-    60          RTS
0343-    A0 00       LDY    #$00
0345-    8C BA 03    STY    $03BA
0348-    C9 A0       CMP    #$A0
034A-    D0 07       BNE    $0353
034C-    A0 80       LDY    #$80
034E-    8C B9 03    STY    $03B9
0351-    D0 D8       BNE    $032B
0353-    A0 00       LDY    #$00
0355-    B9 50 07    LDA    $0750,Y
0358-    20 F0 FD    JSR    $FDF0
035B-    C8          INY
*L

035C-    C0 04       CPY    #$04
035E-    90 F5       BCC    $0355
0360-    A9 A0       LDA    #$A0
0362-    20 F0 FD    JSR    $FDF0
0365-    A0 00       LDY    #$00
0367-    B1 3A       LDA    ($3A),Y
0369-    F0 0C       BEQ    $0377
036B-    CE BB 03    DEC    $03BB
036E-    F0 0D       BEQ    $037D
0370-    2C 00 C0    BIT    $C000
0373-    30 0D       BMI    $0382
0375-    10 20       BPL    $0397
0377-    20 DD FB    JSR    $FBDD
037A-    20 DD FB    JSR    $FBDD
037D-    A0 A0       LDY    #$A0
037F-    8C BB 03    STY    $03BB
0382-    8D 10 C0    STA    $C010
0385-    2C 00 C0    BIT    $C000
0388-    10 FB       BPL    $0385
038A-    AD 00 C0    LDA    $C000
*L
038D-    C9 8D       CMP    #$8D
038F-    F0 06       BEQ    $0397
0391-    C9 A0       CMP    #$A0
0393-    F0 05       BEQ    $039A
0395-    D0 E3       BNE    $037A
0397-    8D 10 C0    STA    $C010
039A-    EA          NOP
039B-    A9 A0       LDA    #$A0
039D-    A0 27       LDY    #$27
039F-    99 50 07    STA    $0750,Y
03A2-    99 D0 07    STA    $07D0,Y
03A5-    88          DEY
03A6-    10 F7       BPL    $039F
03A8-    AD B7 03    LDA    $03B7
03AB-    A0 00       LDY    #$00
03AD-    8C B9 03    STY    $03B9
03B0-    C9 B0       CMP    #$B0
03B2-    90 88       BCC    $033C
03B4-    4C 2B 03    JMP    $032B
03B7-    00          BRK
*L

03B8-    00          BRK
03B9-    00          BRK
03BA-    00          BRK
03BB-    00          BRK
03BC-    2A          ROL
```



自動加插DOS 指令入BASIC 程式中

DOS WRITER

市面上有些應用軟件可以幫助你編寫程式，只要利用這類軟件，編寫程式時便簡易得多。筆者現在要介紹的一個程式，便有類似的用途。

程式的特點是可以將一些 DOS 指令加插入將被更改程式上，例如 CATALOG，LOAD，MENU　等。然而，程式亦可以將一些變數於運算後的結果加入某程式裡。於 EXEC　該程式 FILE 後，便可以取回這些數值作其他用途。

當運行程式時，電腦會詢問你欲要改寫的程式之名字（FILENAME），並從那行句開始。回答問題後，螢幕會打出下列七項的 MENU：

(1) EXEC FILE
(2) EXEC FILE WRITER
(3) STORE VARIABLES
(4) STORE ARRAY
(5) RETRIEVE VARIABLES
(6) RETRIEVE ARRAY
(7) QUIT

選擇「1」時，電腦會將 BASIC 程式加入一個EX－EC指令，使程式 EXEC 一個檔案。

選擇「2」，電腦會自動產生一個 EXEC FILE 。

選擇「3」，或「4」，電腦會從 BASIC 程式內得出的變數數值寫在碟磅內，然後待用。

選擇「5」和「6」，電腦可從磁碟中取得數據，例如「 3」或「4」中得出的數值1寫在程式內一些指定的變數上 。

選擇「7」是結束這程式，同時電腦將你的 BASIC 程式加上上述的指令及 LOAD 入這程式。至於原來的 B－ASIC 程式則放在前面，並加上"ORIGINAL"作爲程式的 FILE NAME，以資識別。此舉是鑑於將來，有問題發生時，仍然可以取回原先的版本。

除了第2項外，在執行各選擇時，電腦都會詢問所輸入的行句是否子程式的末句，如果答Y（是），電腦會在 B－ASIC 程式內自動加上 RETURN，並顯示出子程式的最後一行在那裡。■

DOS WRITER

```
20  TEXT : HOME : PRINT  TAB( 15)
     ;"DOS WRITER": PRINT : PRINT
     : PRINT "PLEASE WAIT A MOMEN
     T"
30 OP$ = "APPEND TEXT FILE":WR$ =
     "WRITE TEXT FILE":CL$ = "CLO
     SE TEXT FILE":D$ =  CHR$ (4)
40  DIM A$(255),A(9,1)
50  PRINT D$;"OPEN TEXT FILE"
60  PRINT D$;"DELETE TEXT FILE"
70  PRINT D$;"OPEN TEXT FILE": PRINT
     D$;"WRITE TEXT FILE": PRINT
     "MON C,I,O": PRINT D$;CL$
80  GOSUB 320
100  REM   MAIN SUBROUTINE
110  HOME : VTAB 2: HTAB 15: PRINT
     "MAIN MENU"
120  HTAB 15: FOR Y = 1 TO 9: PRINT
     "-";: NEXT : PRINT
130  VTAB 6
140  HTAB 15: PRINT "1. EXEC FILE
     "
```

```
150  PRINT
160  HTAB 15: PRINT "2. EXEC FILE
      WRITER"
170  PRINT
180  HTAB 15: PRINT "3. STORE VAR
     IABLES"
190  PRINT
200  HTAB 15: PRINT "4. STORE ARR
     AY"
210  PRINT
220  HTAB 15: PRINT "5. RETRIEVE
     VARIABLES"
230  PRINT
240  HTAB 15: PRINT "6. RETRIEVE
     ARRAY"
250  PRINT
260  HTAB 15: PRINT "7. QUIT"
270  PRINT
280  PRINT : PRINT  TAB( 20)"ENTE
     R ITEM NO. ";: GET A$
290 I% =  VAL (A$): IF I% < 1 OR
     I% > 7 THEN  VTAB 20: GOTO 2
     80
295  PRINT A$
297  IF I% = 2 THEN 420
298  IF I% = 7 THEN 1050
300  IF F1 = 0 THEN  HOME : INPUT
     "STARTING FROM WHICH LINE :-
     ";L:F1 = 1
310  ON I% GOTO 370,420,870,600,1
     020,680,1050
320  HOME : INPUT "NAME OF PROGRA
     M :-";NA$
325  IF  LEN (NA$) = 0 THEN 320
330  PRINT : INPUT "STARTING FROM
      WHICH LINE :-";L:F1 = 1
350  PRINT D$;OP$: PRINT D$;WR$: PRINT
     "NEW": PRINT "SAVE ORIGINAL
     ";NA$: PRINT "DELETE ORIGINA
     L ";NA$: PRINT "RENAME ";NA$
     ;",ORIGINAL ";NA$: PRINT "LO
     AD ORIGINAL ";NA$: PRINT D$;
     CL$: RETURN
370  HOME : INPUT "NAME OF EXEC F
     ILE ? ";NB$
380  PRINT "WHICH DISC DRIVE (1 O
     R 2)?";
390  GET A$: IF A$ <  > "1" AND A
     $ <  > "2" THEN 390
395  PRINT A$
400  PRINT D$;OP$: PRINT D$;WR$: PRINT
     L;"PRINT CHR$(4);"; CHR$ (34
     );"EXEC ";NB$;",D";A$; CHR$
     (34):L = L + 10
410  PRINT D$;CL$: GOTO 1240
420  HOME : INPUT "ENTER NAME OF
     EXEC FILE YOU WISH TO      CR
     EATE :-";NB$
430  IF  LEN (NB$) < 1 THEN 420
460  HOME : PRINT "INPUT COMMANDS
     , TO STOP TYPE ";: INVERSE :
      PRINT "RETURN": NORMAL : PRINT
     "ALONE"
470  FOR I = 0 TO 255: GOSUB 1150
     : IF A$(I) = "" THEN 490
480  NEXT
490  PRINT D$"OPEN";NB$: PRINT D$
     "DELETE";NB$: PRINT D$"OPEN"
     ;NB$: PRINT D$"WRITE";NB$
495 I = I - 1: IF I < 0 THEN 510
500  FOR J = 0 TO I: PRINT A$(J):
      NEXT J
510  PRINT D$"CLOSE";NB$: GOTO 10
     0
600  GOSUB 670
610  PRINT L;"PRINTCHR$(4);"; CHR$
     (34);"DELETE ARRAY ";NB$; CHR$
     (34):L = L + 10
620  PRINT L;"PRINTCHR$(4);"; CHR$
     (34);"OPEN ARRAY ";NB$; CHR$
     (34):L = L + 10
630  PRINT L;"PRINTCHR$(4);"; CHR$
     (34);"WRITE ARRAY ";NB$; CHR$
     (34):L = L + 10
640  FOR I = 1 TO A: PRINT L;"FOR
      I";I;"="A(I,0);" TO ";A(I,1
     ):L = L + 10: NEXT
650  PRINT L;"PRINT"NB$;"(";: FOR
     I = 1 TO A: PRINT "I";I;",";
     : NEXT : PRINT  CHR$ (8);")"
     :L = L + 10
660  GOTO 830
670  REM  SUBROUTINE
680  HOME : INPUT "ENTER THE NAME
      OF THE ARRAY :-";NB$
690  IF  LEN (NB$) < 1 THEN 680
700  INPUT "HOW MANY DIMENSIONS ?
     ";A
710  IF A > 9 THEN  PRINT  CHR$ (
     7);"ARRAY TOO COMPLEX, THE D
     IMENSION CANNOT BE GREATER T
     HAN 9": GOTO 700
720  HOME : PRINT "ENTER THE EXTE
     NT OF THE DIMENSION IN THE F
```

```
    ORM OF < FROM , TO >"
730  POKE 34,2: FOR I = 1 TO A
740  INPUT "";A(I,0),A(I,1): IF A
     (I,0) > A(I,1) THEN  PRINT  CHR$
     (7);"THE FIRST DIMENSION MUS
     T BE SMALLER THANTHE SECOND
     ONE": POKE 34,I + 1: FOR J =
     1 TO 5000: NEXT : HOME : GOTO
     740
750  NEXT : POKE 34,0
760  PRINT D$;OP$: PRINT D$;WR$
770  PRINT L;"REM ARRAY "NB$:L =
     L + 10
780  PRINT L;"PRINTCHR$(4);"; CHR$
     (34);"OPEN ARRAY ";NB$; CHR$
     (34):L = L + 10
790  IF I% = 4 THEN  RETURN
800  PRINT L;"PRINTCHR$(4);" CHR$
     (34);"READ ARRAY ";NB$; CHR$
     (34):L = L + 10
810 : FOR I = 1 TO A: PRINT L;"FO
     R I";I;"=";A(I,0);" TO ";A(I
     ,1):L = L + 10: NEXT
820  PRINT L;"INPUT";NB$"(";: FOR
     I = 1 TO A: PRINT "I";I;",";
     : NEXT : PRINT  CHR$ (8);")"
     :L = L + 10
830  FOR I = 1 TO A: PRINT L;"NEX
     T":L = L + 10: NEXT
840  PRINT L;"PRINTCHR$(4);"; CHR$
     (34);"CLOSE ARRAY ";NB$; CHR$
     (34):L = L + 10
850  PRINT D$;CL$
860  GOTO 1240
870  HOME : INPUT "NAME OF FILE :
      -";NB$
880  IF  LEN (NB$) < 1 THEN 870
890  HOME : PRINT "ENTER NAME OF
     VARIABLE ";: INVERSE : PRINT
     "RETURN";: NORMAL : PRINT "
     ALONE TO FINISH :-"
895  POKE 34,2: FOR I = 1 TO 255
900  INPUT "";A$(I): IF A$(I) = "
     " THEN  GOTO 920
910  NEXT :I = I - 1
920  POKE 34,0:A = I - 1
930  PRINT D$;OP$: PRINT D$;WR$
940  PRINT L;"PRINTCHR$(4);"; CHR$
     (34);"OPEN ";NB$; CHR$ (34):
     L = L + 10: IF I% = 5 THEN  RETURN
950  PRINT L;"PRINTCHR$(4);"; CHR$
     (34);"DELETE ";NB$; CHR$ (34
     ):L = L + 10
960  IF I% = 5 THEN  RETURN
970  PRINT L;"PRINTCHR$(4);"; CHR$
     (34);"OPEN ";NB$; CHR$ (34):
     L = L + 10
980  PRINT L;"PRINTCHR$(4);"; CHR$
     (34);"WRITE ";NB$; CHR$ (34)
     :L = L + 10
990  FOR I = 1 TO A: PRINT L;"PRI
     NT ";A$(I):L = L + 10: NEXT

1000  PRINT L;"PRINTCHR$(4);"; CHR$
     (34);"CLOSE ";NB$; CHR$ (34)
     :L = L + 10
1010  PRINT D$;CL$: GOTO 1240
1020  GOSUB 870
1030  PRINT L;"PRINTCHR$(4);"; CHR$
     (34);"READ ";NB$; CHR$ (34):
     L = L + 10
1040  FOR I = 1 TO A: PRINT L;"IN
     PUT ";A$(I):L = L + 10: NEXT
     : GOTO 1000
1050  HOME : PRINT "PLEASE WAIT A
      MOMENT---"
1060  PRINT D$;OP$: PRINT D$;WR$
1080  PRINT "SAVE ";NA$
1090  PRINT "HOME:LIST"
1100  PRINT "NOMON C,I,O"
1110  PRINT "DELETE TEXT FILE"
1120  PRINT D$;CL$
1130  PRINT D$;"EXEC TEXT FILE"
1140  NEW
1150 A$(I) = "": POKE 51,186: CALL
      - 662
1160  FOR J = 0 TO 255:A =  PEEK
     (512 + J)
1170  IF A = 141 THEN  RETURN
1180 A$(I) = A$(I) +  CHR$ (A): NEXT
     J: RETURN
1240  HOME : PRINT "END OF SUBROU
     TINE (Y/N)? ";: GET A$: IF A
     $ <  > "Y" THEN  PRINT : GOTO
     100
1245  PRINT A$
1248  PRINT "END OF SUBROUTINE AT
      LINE# ";L
1250  PRINT D$;OP$: PRINT D$;WR$
1260  PRINT L;"RETURN":L = L + 10

1270  PRINT D$;CL$:F1 = 0
1280  FOR I = 1 TO 2500: NEXT : GOTO
     100
```

DOS WRITER

以下爲MENU 中各項選擇的使用實例：

一張磁碟內存有的 FILE

```
 A 002 ORIGINAL EXAMPLE
 A 006 EXAMPLE
 A 002 ORIGINAL EXAMPLE 2
 A 005 EXAMPLE 2
 A 004 MENU
*A 016 DOS WRITER
 T 003 ORIGINAL TEXT FILE
 T 002 TEXT FILE
```

(1) LOAD 入 DOS WRITER

(2) 螢幕即會詢問欲加編改程式的 FILENAME。在此例子按入 EXAMPLE，並同時輸入開始的行號，例如 1000。

```
NAME OF PROGRAM :-EXAMPLE
STARTING FROM WHICH LINE :-1000
```

(3) 按入1，並給予新 FILE 一個 FILENAME，例如 EXAMPLE EXEC。

```
                      ENTER ITEM NO. 1
NAME OF EXEC FILE ? EXAMPLE EXEC
WHICH DISC DRIVE (1 OR 2)?1
END OF SUBROUTINE (Y/N)?
```

(4) 按入選擇2，
螢幕會要求你輸入 EXEC FILE 的名字（例如 EXAMPLE EXEC）然後輸入欲加的 DOS 指令。假如 CATALOG。

```
                      ENTER ITEM NO. 2
ENTER NAME OF EXEC FILE YOU WISH TO
CREATE :-EXEB EXAMPLE EXEC
INPUT COMMANDS, TO STOP TYPE RETURN
ALONE
:CATALOG
:
```

(5) 按入選擇3，
輸入變數名稱：例如A，Z，A$，Z$，N%，A。（在未輸入前，先回答 FILE 名字，例如 V－ARIABLES。

```
                      ENTER ITEM NO. 3
NAME OF FILE : -VARIABLES
ENTER NAME OF VARIABLE RETURN ALONE
TO FINISH :-   A
               Z
               A$
               Z$
               N%
```

(6) 按入選擇4，
螢幕會要求輸入陣列（ARRAY）名字，例如－A%－B，－C$等同時亦要輸入陣列的 DIMENSION。

```
                      ENTER ITEM NO. 4
ENTER THE NAME OF THE ARRAY :-A%
HOW MANY DIMENSIONS ?2
ENTER THE EXTENT OF THE DIMENSION
IN THE FORM OF < FROM , TO >
1,5
1,4
END OF SUBROUTINE (Y/N)?

                      ENTER ITEM NO. 4
ENTER THE NAME OF THE ARRAY :-B
HOW MANY DIMENSIONS ?2
ENTER THE EXTENT OF THE DIMENSION
IN THE FORM OF < FROM , TO >
1,3
1,2
END OF SUBROUTINE (Y/N)?

                      ENTER ITEM NO. 4
ENTER THE NAME OF THE ARRAY :-C$
HOW MANY DIMENSIONS ?1
ENTER THE EXTENT OF THE DIMENSION
IN THE FORM OF < FROM , TO >
1,26
END OF SUBROUTINE (Y/N)? Y
END OF SUBROUTINE AT LINE# 1430
```

(7) 之後，按入7，然後等一會兒。

(8) 下例爲總括電腦執行過程，由此而會產生一個新 FILE。

```
]NEW

]SAVE ORIGINAL EXAMPLE

]DELETE ORIGINAL EXAMPLE

]RENAME EXAMPLE,ORIGINAL EXAMPLE

]LOAD ORIGINAL EXAMPLE
]1000PRINT CHR$(4);"EXEC EXAMPLE EXEC,D1"

]1010PRINTCHR$(4);"OPEN VARIABLES"

]1020PRINTCHR$(4);"DELETE VARIABLES"

]1030PRINTCHR$(4);"OPEN VARIABLES"

]1040PRINTCHR$(4);"WRITE VARIABLES"

]1050PRINT A

]1060PRINT Z

]1070PRINT A$

]1080PRINT Z$

]1090PRINT N%

]1100PRINT A

]1110PRINTCHR$(4);"CLOSE VARIABLES"

]1120REM ARRAY A%

]1130PRINTCHR$(4);"OPEN ARRAY A%"

]1140PRINTCHR$(4);"DELETE ARRAY A%"

]1150PRINTCHR$(4);"OPEN ARRAY A%"

]1160PRINTCHR$(4);"WRITE ARRAY A%"

]1170FOR I1=1 TO 5

]1180FOR I2=1 TO 4

]1190PRINTA%(I1,I2)

]1200NEXT

]1210NEXT

]1220PRINTCHR$(4);"CLOSE ARRAY A%"

]1230REM ARRAY B

]1240PRINTCHR$(4);"OPEN ARRAY B"

]1250PRINTCHR$(4);"DELETE ARRAY B"

]1260PRINTCHR$(4);"OPEN ARRAY B"

]1270PRINTCHR$(4);"WRITE ARRAY B"

]1280FOR I1=1 TO 3

]1290FOR I2=1 TO 2

]1300PRINTB(I1,I2)

]1310NEXT

]1320NEXT

]1330PRINTCHR$(4);"CLOSE ARRAY B"

]1340REM ARRAY C$

]1350PRINTCHR$(4);"OPEN ARRAY C$"

]1360PRINTCHR$(4);"DELETE ARRAY C$"

]1370PRINTCHR$(4);"OPEN ARRAY C$"

]1380PRINTCHR$(4);"WRITE ARRAY C$"

]1390FOR I1=1 TO 26

]1400PRINTC$(I1)

]1410NEXT

]1420PRINTCHR$(4);"CLOSE ARRAY C$"

]1430RETURN

]SAVE EXAMPLE
```

(9) 新 FILE 產生之後，電腦會將新 FILE 重新列表一次。

```
]HOME:LIST

10  REM  THIS TESTS OPTIONS 1-4
20 A = 1:Z = 26:A$ = "HELLO":Z$ =
     "GOODBYE":N% = 14:VA = 12.34

30  DIM A%(5,4),B(3,2),C$(26)
40  FOR I = 1 TO 5: FOR J = 1 TO
     4:A%(I,J) = 5 * I + J: NEXT
     : NEXT
50  FOR I = 1 TO 3: FOR J = 1 TO
     2:B(I,J) = 1.1 * I + J: NEXT
     : NEXT
60  FOR I = 1 TO 26:C$(I) =  STR$
     (I): NEXT
70  GOSUB 1000
80  END
160  PRINT "CATALOG"
1000  PRINT  CHR$ (4);"EXEC EXAMP
     LE EXEC,D1"
1010  PRINT  CHR$ (4);"OPEN VARIA
     BLES"
1020  PRINT  CHR$ (4);"DELETE VAR
     IABLES"
1030  PRINT  CHR$ (4);"OPEN VARIA
     BLES"
1040  PRINT  CHR$ (4);"WRITE VARI
     ABLES"
1050  PRINT A
1060  PRINT Z
1070  PRINT A$
1080  PRINT Z$
1090  PRINT N%
1100  PRINT A
1110  PRINT  CHR$ (4);"CLOSE VARI
     ABLES"
1120  REM  ARRAY A%
1130  PRINT  CHR$ (4);"OPEN ARRAY
     A%"
1140  PRINT  CHR$ (4);"DELETE ARR
     AY A%"
1150  PRINT  CHR$ (4);"OPEN ARRAY
     A%"
1160  PRINT  CHR$ (4);"WRITE ARRA
     Y A%"
1170  FOR I1 = 1 TO 5
1180  FOR I2 = 1 TO 4
1190  PRINT A%(I1,I2)
1200  NEXT
1210  NEXT
1220  PRINT  CHR$ (4);"CLOSE ARRA
     Y A%"
1230  REM  ARRAY B
1240  PRINT  CHR$ (4);"OPEN ARRAY
     B"
1250  PRINT  CHR$ (4);"DELETE ARR
     AY B"
1260  PRINT  CHR$ (4);"OPEN ARRAY
     B"
1270  PRINT  CHR$ (4);"WRITE ARRA
     Y B"
1280  FOR I1 = 1 TO 3
1290  FOR I2 = 1 TO 2
1300  PRINT B(I1,I2)
1310  NEXT
1320  NEXT
1330  PRINT  CHR$ (4);"CLOSE ARRA
     Y B"
1340  REM  ARRAY C$
1350  PRINT  CHR$ (4);"OPEN ARRAY
     C$"
1360  PRINT  CHR$ (4);"DELETE ARR
     AY C$"
1370  PRINT  CHR$ (4);"OPEN ARRAY
     C$"
1380  PRINT  CHR$ (4);"WRITE ARRA
     Y C$"
1390  FOR I1 = 1 TO 26
1400  PRINT C$(I1)
1410  NEXT
1420  PRINT  CHR$ (4);"CLOSE ARRA
     Y C$"
1430  RETURN
1440  PRINT  CHR$ (4);"OPEN ARRAY
     C$"
1450  PRINT  CHR$ (4);"WRITE ARRA
     Y C$"
1460  FOR I1 = 1 TO 26
1470  PRINT C$(I1)
1480  NEXT
1490  PRINT  CHR$ (4);"CLOSE ARRA
     Y C$"
1500  RETURN

]NOMON C,I,O
```

以下為選擇6的應用實例：

(1) LOAD 入 DOS WRITER 後，輸入 EXAMPLE 2 作程式 FILENAME。並決定那行句開始（此例子中同樣為 1000）。

```
NAME OF PROGRAM :-EXAMPLE 2

STARTING FROM WHICH LINE :-1000
```

(2) 按入選擇6，輸入陣列的名字及其 DIMENSION。例如—Y，DIMENSION；—Z$，DIMENSION 1，—X% DIMENSION 2 等。

```
                    ENTER ITEM NO. 6
ENTER THE NAME OF THE ARRAY :-X%
HOW MANY DIMENSIONS ?2
ENTER THE EXTENT OF THE DIMENSION
IN THE FORM OF < FROM , TO >
1,5
1,4
END OF SUBROUTINE (Y/N)?
                    ENTER ITEM NO. 6
ENTER THE NAME OF THE ARRAY :-Y
HOW MANY DIMENSIONS ?2
ENTER THE EXTENT OF THE DIMENSION
IN THE FORM OF < FROM , TO >
1,3
1,2
END OF SUBROUTINE (Y/N)?

                    ENTER ITEM NO. 6
ENTER THE NAME OF THE ARRAY :-Z$
HOW MANY DIMENSIONS ?1
ENTER THE EXTENT OF THE DIMENSION
IN THE FORM OF < FROM , TO >
1,26
END OF SUBROUTINE (Y/N)? Y
END OF SUBROUTINE AT LINE# 1250
```

(3) 一切預備後，電腦會詢問子網路是否完結，答是（Y）。

(4) 按入選擇7，終結程式並等候一會。以下為電腦顯示其執行過程：

```
]NEW

]SAVE ORIGINAL EXAMPLE 2

]DELETE ORIGINAL EXAMPLE 2

]RENAME EXAMPLE 2,ORIGINAL EXAMPLE 2

]LOAD ORIGINAL EXAMPLE 2
]1000REM ARRAY X%

]1010PRINTCHR$(4);"OPEN ARRAY X%"

]1020PRINTCHR$(4);"READ ARRAY X%"

]1030FOR I1=1 TO 5

]1040FOR I2=1 TO 4

]1050INPUTX%(I1,I2)

]1060NEXT

]1070NEXT

]1080PRINTCHR$(4);"CLOSE ARRAY X%"

]1090REM ARRAY Y

]1100PRINTCHR$(4);"OPEN ARRAY Y"

]1110PRINTCHR$(4);"READ ARRAY Y"

]1120FOR I1=1 TO 3

]1130FOR I2=1 TO 2

]1140INPUTY(I1,I2)

]1150NEXT

]1160NEXT

]1170PRINTCHR$(4);"CLOSE ARRAY Y"

]1180REM ARRAY Z$

]1190PRINTCHR$(4);"OPEN ARRAY Z$"

]1200PRINTCHR$(4);"READ ARRAY Z$"

]1210FOR I1=1 TO 26

]1220INPUTZ$(I1)

]1230NEXT

]1240PRINTCHR$(4);"CLOSE ARRAY Z$"

]1250RETURN

]SAVE EXAMPLE 2
```

5) 電腦 將新 FILE 重新列表。

```
]HOME:LIST

10  REM  THIS TESTS OPTIONS 5-6
20 X = 0:Y = 0:X$ = "":Y$ = "":Z%
     = 0:ZZ = 0
30  DIM X%(5,4),Y(3,2),Z$(26)
40  GOSUB 1000
50  PRINT "1=";X: PRINT "26=";Y: PRINT
     "HELLO=";X$: PRINT "GOODBYE=
     ";Y$: PRINT "14=";Z%: PRINT
     "12.34=";ZZ
60  PRINT "23=";X%(4,3): PRINT "3
     .2=";Y(2,1): PRINT "20=";Z$(
     20)
70  END
1000  REM  ARRAY X%
1010  PRINT  CHR$ (4);"OPEN ARRAY
       X%"
1020  PRINT  CHR$ (4);"READ ARRAY
       X%"
1030  FOR I1 = 1 TO 5
1040  FOR I2 = 1 TO 4
1050  INPUT X%(I1,I2)
1060  NEXT
1070  NEXT
1080  PRINT  CHR$ (4);"CLOSE ARRA
     Y X%"
1090  REM  ARRAY Y
1100  PRINT  CHR$ (4);"OPEN ARRAY
       Y"
1110  PRINT  CHR$ (4);"READ ARRAY
       Y"
1120  FOR I1 = 1 TO 3
1130  FOR I2 = 1 TO 2
1140  INPUT Y(I1,I2)
1150  NEXT
1160  NEXT
1170  PRINT  CHR$ (4);"CLOSE ARRA
     Y Y"
1180  REM  ARRAY Z$
1190  PRINT  CHR$ (4);"OPEN ARRAY
       Z$"
1200  PRINT  CHR$ (4);"READ ARRAY
       Z$"
1210  FOR I1 = 1 TO 26
1220  INPUT Z$(I1)
1230  NEXT
1240  PRINT  CHR$ (4);"CLOSE ARRA
     Y Z$"
1250  RETURN
1260  PRINT  CHR$ (4);"CLOSE ARRA
     Y Y"
1270  REM  ARRAY Z$
1280  PRINT  CHR$ (4);"OPEN ARRAY
       Z$"
1290  PRINT  CHR$ (4);"READ ARRAY
       Z$"
1300  FOR I1 = 1 TO 26
1310  INPUT Z$(I1)
1320  NEXT
1330  PRINT  CHR$ (4);"CLOSE ARRA
     Y Z$"
1340  RETURN

]NOMON C,I,O

]
```

TEXT FILE 有些什麼內容？

本程式將忘懷了的TEXT FILE 內容重新讀出

我們經常忘記許多 TEXT FILE 的內容，或者更甚的，根本忘記了某些 TEXT FILE 的存在。許多時有些 TEXT FILE 因爲忘記了其長度，形式而無法恢復？要解決這些困擾相當容易，本文提供一個非常實用的程式給大家。

本文附載的程式可以讀出任何磁碟中的 TEXT FILE，從這程式中，可知道該 TEXT FILE 是 SQUENT—IAL FILE 或是 RANDOM ACCESS FILE，並可進一步知道 FILE 共有多少個 FIELD 及 RECORD。

當 RUN 本程式後，你便可將要讀的 TEXT FILE 名字打入，程式便會從第二部磁碟機（DRIVE 2）找出。你更可以將資料從印字機中印出。如果你只得一部磁碟機，可將程式中的行句 515 中 DR＝2 改爲 DR＝1。或第一次入資料時，只按 RETURN，程式便可將其中一個磁碟機內磁碟上的 TEXT FILE 列出。

當電腦列出 TEXT FILE 內容時，只要按任何鍵都可暫停印出 FILE 內容，再按一次列出會繼續。

按 ESC 便會停止列出 TEXT FILE 的內容。

本程式乃以 APPLESOFT BASIC 寫成，所以執行時速度較慢，有時需耐心等候。（注意，有時要等候超過一分鐘或以上）。由於有部份 DOS 子程式被修改過，因此當程式完結前必須將所改變了的 DOS 子程式回復原狀。如中途按 RESET，是不能輸入 DOS 指令，必須重新 R—EBOOT或打入 GOTO 5000。

有關 TEXT FILE 的結構，讀者可參閱本刊第一期的 DOS 專欄。

本程式分爲三個主要部份：

行號 510 至 710 是找尋子網路，專責找尋在 CA—TALOG中的 TEXT FILE。

行號 810 至1010是閱讀 FILE 之用。

行號1210至1320是結束網路，用以給出有 OPEN TEXT FILE 的資料。

全個程式依賴始於 $ B037 位址的DOS 網路。DOS 利用這些網路來寫出磁碟的 DIRECTORY 在磁碟上，但我們却可以利用它來讀及寫任何 SECTOR，無論是從磁碟中或往磁碟去均可。我們只須在讀入時把 1 放在 $B041上，而在寫出時，把 2 寫在 $B041 上，同時須把處理的 TRACK 數目寫在 $B397上，SECTOR 數目寫在 $B398上。而 DOS 亦同時讀入（或寫出）有關 SECT—OR 往（或從）$B4BB—$B5BA的FILE 緩衝器上。

因此本程式將這些工作簡化爲先 POKE 入須處理的 TRACK及 SECTOR及讀入／寫出資料在正確的位址上，再 CALL 45111 便完成了。□

檔案內容讀出程式

```
10   POKE 40286,35: POKE 40287,216
20   ONERR  GOTO 5000
30   GOTO 510
100   HTAB 20 -  LEN (Z$) / 2: INVERSE
     : PRINT Z$: NORMAL : RETURN
150   IF D$ = "P" THEN  PR# 1
160   RETURN
180   IF  NOT FL THEN NG = NG + 1
190   RETURN
200   IF KB = 155 THEN  GOSUB 150:
      HTAB X + 1: VTAB Y + 1: PRINT
     : PRINT : INVERSE : PRINT "S
     CANNING ABORTED !": NORMAL :
      GOSUB 180: PRINT : PRINT "A
     CCORDING TO THE INFORMATION
     WE HAVE": POP : GOTO 1210
205   PR# 0
210 KB =  PEEK ( - 16384): IF KB >
     127 AND KB <  > 155 THEN  POKE
      - 16368,0: VTAB 3:Z$ = "ANY
      OTHER KEY TO STOP SCROLLING
     ": GOSUB 100: HTAB X + 1: VTAB
     Y + 1: RETURN
220   VTAB 3:Z$ = "   ANY OTHER KE
     Y TO CONTINUE   ": GOSUB 100
230   GOTO 200
510 RW = 45121:TK = 45975:SC = 45
     976:R = 1:W = 2:SB = 46267
515 SL = 6 * 16:DR = 2: POKE 4658
     3,SL: POKE 46584,DR
520   HOME : VTAB 5
530   PRINT "  THIS PROGRAM WILL A
     LLOW YOU TO FIND": PRINT "
     OUT WHAT'S ON YOUR TEXT FILE
     S.": PRINT "  ALL YOU HAVE T
     O DO IS ENTER THE": PRINT "
      FILENAME AND WAIT. IT WILL
     PRINT OUT": PRINT "  ALL THA
     T'S IN,WHETHER ACTUAL CONTEN
     T"
531   PRINT "  OR OLD OVERWRITTEN
     STUFF."
540   VTAB 14: PRINT "  SO,WHAT'S
     YOUR FILE'S NAME ?"
542   VTAB 22: PRINT "   PRESS ";:
      INVERSE : PRINT "RETURN";:
     NORMAL : PRINT " ONLY TO
     CATALOG"
543   VTAB 16: INPUT " ";A$:A =  LEN
     (A$): IF A > 30 THEN  GOSUB
     1500: VTAB 16: CALL  - 868: GOTO
```

```
540
545  IF A = 0 THEN 5500
550  HTAB 1: VTAB 18: PRINT " SCR
     EEN OR PRINTER OUTPUT ? (S/P
     ) ";: GET D$: IF D$ <  > "S"
      AND D$ <  > "P" THEN 550
555 A$ =  LEFT$ (A$ + "
                          ",30)
560  PRINT : VTAB 2:Z$ = "HIT ESC
      TO STOP WORK ": GOSUB 100:Z
     $ = "ANY OTHER KEY TO STOP S
     CROLLING": GOSUB 100: POKE 3
     4,3
570  HOME
600  FOR I = 15 TO 1 STEP  - 1: VTAB
     1: CALL  - 868:Z$ = "LOOKING
      FOR FILE IN CATALOG; SECTOR
      " +  STR$ (I): GOSUB 100: POKE
     RW,R: POKE TK,17: POKE SC,I:
      CALL 45111
610  FOR J = 0 TO 6:CK = (SB + 11
      + 35 * J)
620  IF  PEEK (CK) = 0 AND  PEEK
     (CK + 1) = 0 THEN  INVERSE :
      VTAB 22: PRINT "THERE'S NO
     SUCH FILE ON THIS DISKETTE !
     ": NORMAL : GOTO 5010
630  IF  PEEK (CK) = 255 THEN 710

640 F$ = ""
650  FOR K = 0 TO 29:F$ = F$ +  CHR$
     ( PEEK (CK + 3 + K) - 128 *
      INT ( PEEK (CK + 3 + K) / 1
     28)): NEXT K
660  IF A$ <  > F$ THEN 710
670  IF  PEEK (CK + 2) <  > 0 AND
      PEEK (CK + 2) <  > 128 THEN
      INVERSE : VTAB 22: PRINT "
      SORRY,THIS IS NOT A TEXT FI
     LE        "; NORMAL : GOTO 50
     10
680 TTSL =  PEEK (CK):STSL =  PEEK
     (CK + 1)
700 J = 6:I = 1
710  NEXT J: NEXT I
810  DIM T(122),S(122):MF = 10000
     :MIN = 10000:SP =  - 1:X = 0
     :Y = 3
820  IF TTSL = 0 AND STSL = 0 THEN
     1200
825  VTAB 1: CALL  - 868:Z$ = "RE
     ADING TRACK/SECTOR LIST": GOSUB
     100: POKE TK,TTSL: POKE SC,S
     TSL: POKE RW,R: CALL 45111
840 TTSL =  PEEK (SB + 1):STSL =
      PEEK (SB + 2)
850  FOR I = 1 TO 122:T(I) =  PEEK
     (SB + 10 + 2 * I):S(I) =  PEEK
     (SB + 11 + 2 * I): NEXT I
860  FOR I = 1 TO 122: IF T(I) =
     0 AND S(I) = 0 THEN 1200
870  VTAB 1: CALL  - 868:Z$ = " R
     EADING SECTOR " +  STR$ (LN /
     256 + 1) + " OF FILE": GOSUB
     100: POKE TK,T(I): POKE SC,S
     (I): POKE RW,R: CALL 45111
880  POKE  - 16368,0
890  VTAB 1: CALL  - 868:Z$ = "SC
     ANNING SECTOR " +  STR$ (LN /
     256 + 1) + " OF FILE": GOSUB
     100: GOSUB 150: VTAB Y + 1: HTAB
     X + 1: FOR J = 0 TO 255:C =
      PEEK (SB + J):C$ =  CHR$ (C
     ):LN = LN + 1:SP = SP + 1
900 KB =  PEEK ( - 16384): IF KB >
     127 THEN  POKE  - 16368,0:X =
      PEEK (36):Y =  PEEK (37): GOSUB
     200: GOSUB 150
905  IF (C = 44 OR C = 172) AND  NOT
     FC THEN  GOSUB 1700
910  IF ((C = 44 OR C = 172) AND
     CF) OR C = 13 OR C = 141 THEN
     NG = NG + 1:NF = NF + 1
930 FP = FL:FL =  NOT C: IF FP =
     0 THEN 1000
940  IF FL THEN ZR = ZR + 1: GOTO
     1000
950 ZR = 0:NR = NR + 1: INVERSE :
      PRINT : PRINT "THIS IS THE
     END OF A RECORD !!": NORMAL
     : IF SP > MAX THEN MAX = SP
960  IF SP < (MIN) THEN MIN = SP
970 SP = 0
980  IF NR > 1 AND NF THEN  IF NF
      < MF THEN :MF = NF
990 NF = 0
1000  IF D$ = "S" THEN  PRINT C$;
     : IF (C = 44 OR C = 172) AND
     CF THEN  PRINT
1002  IF D$ = "P" THEN ZZ$ = ZZ$ +
     C$: IF C = 13 OR C = 141 OR
     ((C = 44 OR C = 172) AND CF)
      THEN  PRINT ZZ$;:ZZ$ = "": IF
     C = 44 OR C = 172 THEN  PRINT

1003  IF  LEN (ZZ$) = 79 THEN  PRINT
     ZZ$:ZZ$ = ""
1005  NEXT J:X =  PEEK (36):Y =  PEEK
```

TEXT FILE 有些什麼內容？

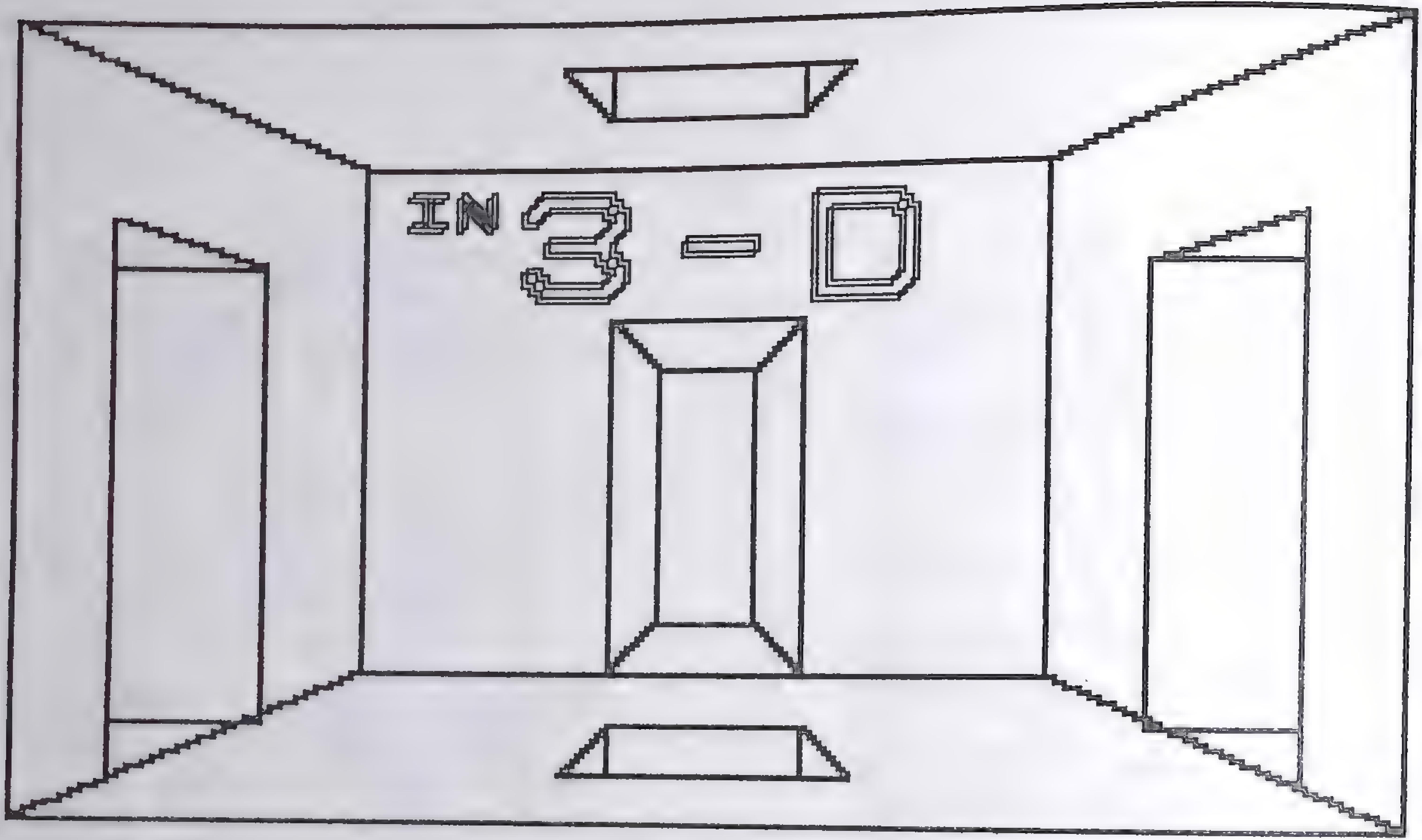
```
     (37): PR# 0: NEXT
1010  GOTO 820
1200  PRINT : INVERSE : PRINT "TH
     IS IS THE END OF THE FILE":
     NORMAL : PRINT
1210 TY$ = "RANDOM ACCESS": IF NR
      = 0 THEN TY$ = "SEQUENTIAL"

1220  GOSUB 1600: IF D < 5 THEN T
     Y$ = "PROBABLY SEQ.": IF NR >
     2 THEN TY$ = TY$ + ",MAYBE R
     EDEF. R.A."
1230  PRINT "THE MOST POSSIBLE CH
     ARACTERISTICS": PRINT "OF YO
     UR FILE ARE FOLLOWING:"
1240  PRINT "TYPE :";TY$
1250  PRINT "LENGTH :";:LE = ( INT
     ((LN - ZR) / MIN) + 1) * MIN
     : IF  LEFT$ (TY$,1) = "S" OR
      LEFT$ (TY$,1) = "P" THEN LE
      = LN - ZR
1260  PRINT LE;" BYTES"
1270  PRINT "NB.OF FIELDS ";: IF
     TY$ = "RANDOM ACCESS" THEN  PRINT
     "PER RECORD ";
1280 NB = NG: IF TY$ = "RANDOM AC
     CESS" THEN NB = (MF < 10000)
      * MF
1290  PRINT NB
1300  IF TY$ = "RANDOM ACCESS" THEN
      PRINT "NB.OF RECORDS :";LE /
     MIN: PRINT "NB.OF ACTUALLY D
     EFINED RECORDS :";NR + 1: PRINT
     "RECORD LENGTH (PROBABLE):";
     MIN
1310  PRINT : PRINT : PRINT
1320  PR# 0: GOTO 5010
1500  VTAB 22: INVERSE : PRINT "N
     OT MORE THAN 30 CHARACTERS P
     LEASE !!": NORMAL : FOR X =
     1 TO 300:Y =  PEEK ( - 16336
     ): NEXT : VTAB 22: CALL  - 8
     68: RETURN
1600 A = MAX:B = MIN:D = 1
1610 C = B: IF C = 1 THEN  RETURN

1620  FOR L = 2 TO C: IF A / L =
     INT (A / L) AND B / L =  INT
     (B / L) THEN A = A / L:B = B
     / L:D = D * L:L = C: NEXT L
     : GOTO 1610
1630  NEXT L: RETURN
1700  PR# 0:X =  PEEK (36):Y =  PEEK
     (37): VTAB 22: HTAB 1: PRINT
     "COMMAS AS FIELD SEPARATORS
     ? ";: GET Y$: IF Y$ <  > "Y"
      AND Y$ <  > "N" THEN 1700
1710  VTAB 22: HTAB 1: CALL  - 86
     8: VTAB Y + 1: HTAB X + 1: GOSUB
     150
1720  IF Y$ = "Y" THEN CF = 1
1730 FC = 1: RETURN
5000 ER =  PEEK (222): IF ER <  >
     16 AND ER <  > 255 THEN  VTAB
     21: HTAB 1: FLASH : PRINT  CHR$
     (7): PRINT "ERROR NR. "ERR"
     IN LINE " PEEK (218) +  PEEK
     (219) * 256
5010  POKE RW,W: VTAB 23: HTAB 1:
      CALL  - 868: INVERSE : PRINT
     "EXITING PROGRAM, DOS RESTOR
     ED": NORMAL : POKE 34,0: POKE
      - 16368,0: END
5500  HOME : VTAB 10: PRINT "ON W
     HICH DRIVE (1/2)?";: GET A$:
     DR =  VAL (A$)
5510  IF DR <  > 1 AND DR <  > 2 THEN
     5500
5520  HOME : PRINT "TEXT FILES ON
      DRIVE ";DR
5523  POKE 34,1
5525  POKE 46584,DR
5530  GOSUB 6000
5540  PRINT : VTAB 23: PRINT "PRE
     SS ANY KEY TO CONTINUE"
5550 KB =  PEEK ( - 16384): IF KB
      < 128 THEN 5550
5555  POKE 34,0
5560  POKE  - 16368,0: HOME : GOTO
     540
6000  FOR I = 15 TO 1 STEP  - 1
6010  POKE RW,R: POKE TK,17: POKE
     SC,I: CALL 45111
6020  FOR J = 0 TO 6:CK = (SB + 1
     1 + 35 * J)
6030  IF  PEEK (CK) = 0 AND  PEEK
     (CK + 1) = 0 THEN  PRINT : RETURN

6040  IF  PEEK (CK) = 255 THEN 60
     70
6050  IF  PEEK (CK + 2) <  > 0 AND
      PEEK (CK + 2) <  > 128 THEN
     6070
6060  FOR K = 0 TO 29: PRINT  CHR$
     ( PEEK (CK + 3 + K) - 128 *
      INT ( PEEK (CK + 3 + K) / 1
     28));: NEXT K: PRINT
6070  NEXT J: NEXT I: RETURN
```

立體迷宮



當遊戲開始，你便會處身一個5×5立體迷宮裡面，你可從電視螢光屏上看到上、下、前、左、右五個方向，可能其中有一些方向會有通道，你要盡快沿這些通道離這個迷宮。

在電視螢光屏左下方，會顯視出進入迷宮後的時間，其右方則顯示你所面對的方向。最低一行三欄數字是顯示所處的方位。

你可被“F”來改變所面對的方向，電腦會問所對爲何方向，玩者可任意選擇N—北，S—南，E—東或W—西。

如欲穿過某方的門，你可按U—上，D—下，N—北，S—南，E—東或W—西。當穿過一度門，你會發現你處身在另一個房間。

按“☆”時，電腦便會顯示迷宮的出口座標。

按“?”時，電腦便會顯示你所在的位置座標。

按“Q”時，遊戲便終止，你可以將未完成的迷宮貯入磁碟內，下次你 RUN 這遊戲時可以繼續利用這迷宮。

座標由三個數目表示：

左邊數值表示東西方向（向東走數值加一）數值在1至5範圍。

中間數值表示南北（向南行數值加一）

右邊數值表示上下（向上走數值加一）

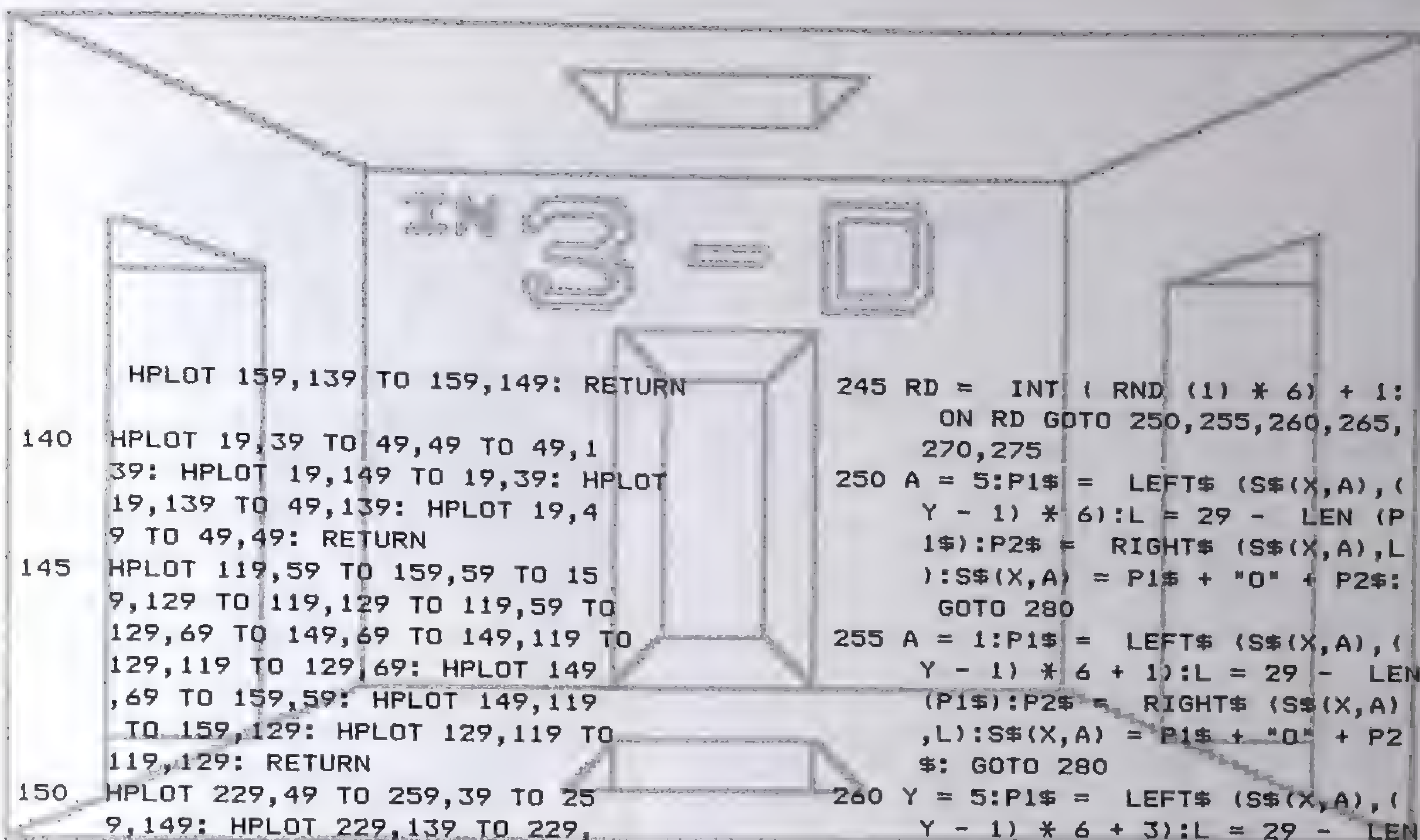
假如螢幕上3個數值與答案所顯示出的數值相同，即表示已成功走出迷宮。

當電腦在建立迷宮時，需等候一分鐘，請稍爲忍耐！■

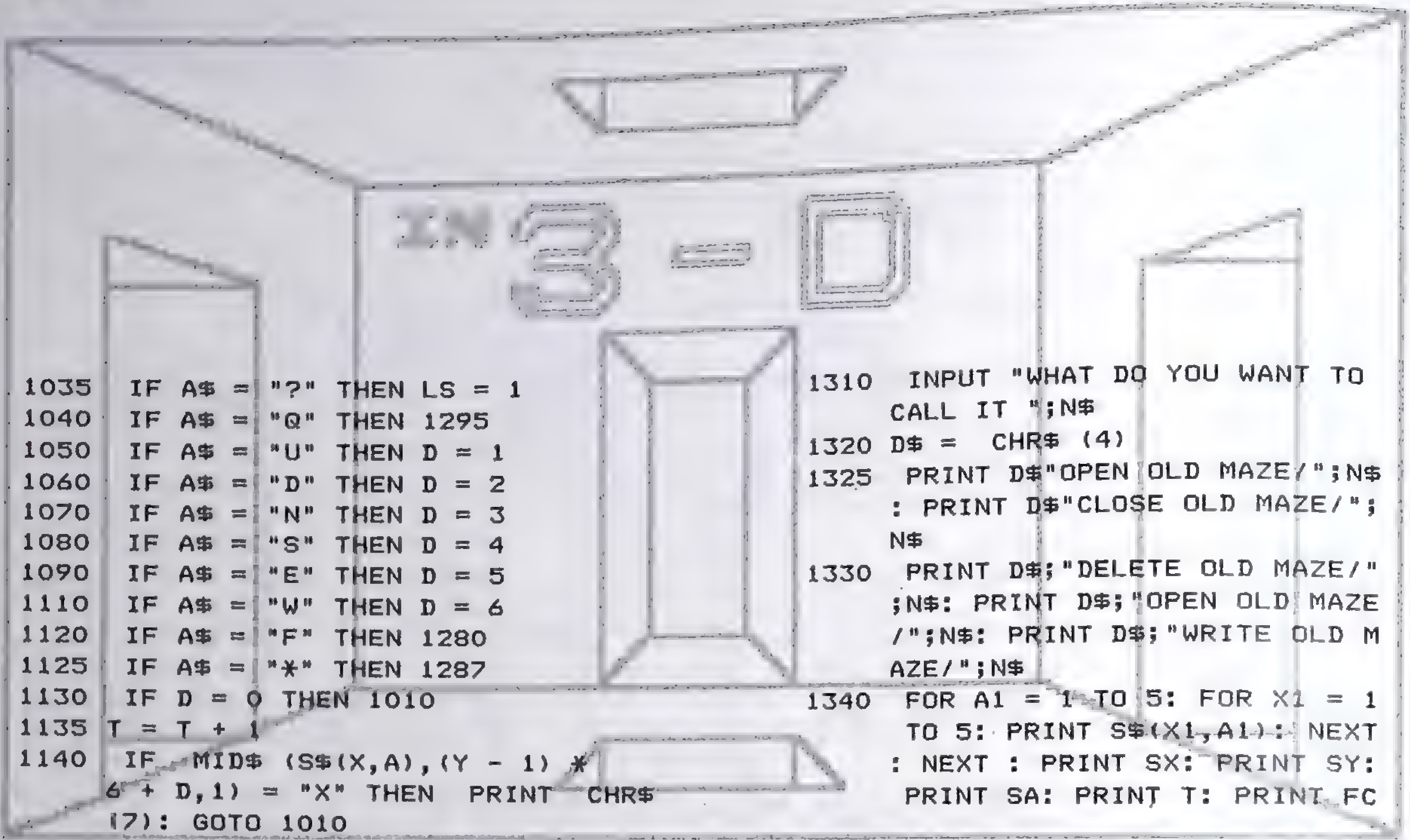
立體迷宮程式

```
10  TEXT : HOME
90  GOSUB 2000
100  DIM FC(5,7): DIM FC$(5)
105 FC$(1) = "NORTH":FC$(2) = "SO
    UTH":FC$(3) = "EAST":FC$(4) =
    "WEST"
110  FOR B = 1 TO 4: FOR I = 1 TO
    6: READ FC(B,I): NEXT : NEXT

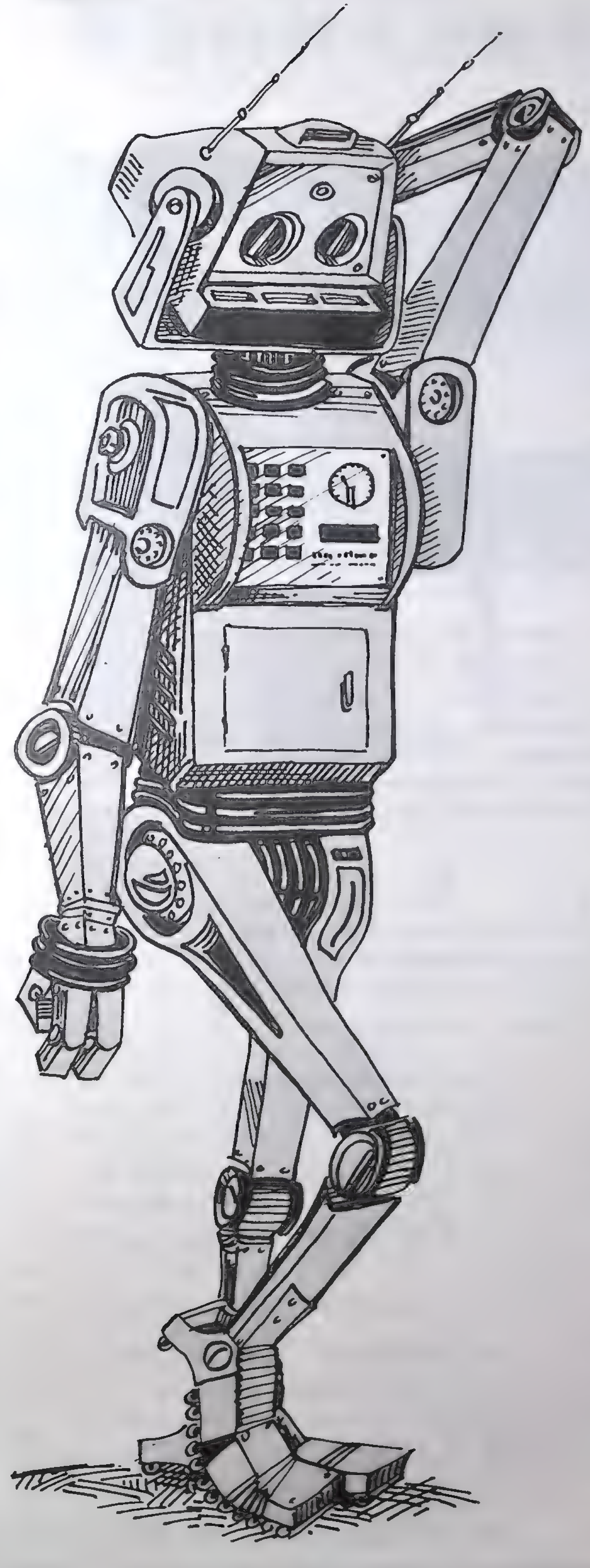
115  GOTO 155
120  HPLOT 0,0 TO 279,0 TO 279,15
    9 TO 0,159 TO 0,0 TO 69,29 TO
    209,29 TO 209,129 TO 69,129 TO
    69,29: HPLOT 209,29 TO 279,0
    : HPLOT 209,129 TO 279,159: HPLOT
    69,129 TO 0,159: RETURN
125  RETURN
130  HPLOT 109,9 TO 169,9 TO 159,
    19 TO 119,19 TO 109,9: HPLOT
    119,19 TO 119,9: HPLOT 159,1
    9 TO 159,9: RETURN
135  HPLOT 119,139 TO 159,139 TO
    169,149 TO 109,149 TO 119,13
    9: HPLOT 119,139 TO 119,149:
```




```
 HPLOT 159,139 TO 159,149: RETURN
140  HPLOT 19,39 TO 49,49 TO 49,1
     39: HPLOT 19,149 TO 19,39: HPLOT
     19,139 TO 49,139: HPLOT 19,4
     9 TO 49,49: RETURN
145  HPLOT 119,59 TO 159,59 TO 15
     9,129 TO 119,129 TO 119,59 TO
     129,69 TO 149,69 TO 149,119 TO
     129,119 TO 129,69: HPLOT 149
     ,69 TO 159,59: HPLOT 149,119
      TO 159,129: HPLOT 129,119 TO
     119,129: RETURN
150  HPLOT 229,49 TO 259,39 TO 25
     9,149: HPLOT 229,139 TO 229,
     49: HPLOT 229,49 TO 259,49: HPLOT
     229,139 TO 259,139: RETURN
155  DIM S$(6,6)
160  INPUT "RESTART OLD MAZE ";Y$
     : IF  LEFT$ (Y$,1) = "Y" THEN
     1360
163  GOSUB 1500
165  FOR A = 1 TO 5: FOR X = 1 TO
     5: FOR Y = 1 TO 5
170  IF A <  > 5 AND  RND (1) < .
     80 THEN S$(X,A) = S$(X,A) +
     "O": GOTO 180
175 S$(X,A) = S$(X,A) + "X"
180  IF  MID$ (S$(X,A - 1),(Y - 1
     ) * 6 + 1,1) = "O" THEN S$(X
     ,A) = S$(X,A) + "O": GOTO 19
     0
185 S$(X,A) = S$(X,A) + "X"
190  IF Y - 2 < 0 THEN 200
195  IF  MID$ (S$(X,A),(Y - 2) *
     6 + 4,1) = "O" THEN S$(X,A) =
     S$(X,A) + "O": GOTO 205
200 S$(X,A) = S$(X,A) + "X"
205  IF Y <  > 5 AND  RND (1) < .
     8 THEN S$(X,A) = S$(X,A) + "
     O": GOTO 215
210 S$(X,A) = S$(X,A) + "X"
215  IF X <  > 5 AND  RND (1) < .
     8 THEN S$(X,A) = S$(X,A) + "
     O": GOTO 225
220 S$(X,A) = S$(X,A) + "X"
225  IF  MID$ (S$(X - 1,A),(Y - 1
     ) * 6 + 5,1) = "O" THEN S$(X
     ,A) = S$(X,A) + "O": GOTO 23
     5
230 S$(X,A) = S$(X,A) + "X"
235  NEXT : NEXT : NEXT
240 X =  INT ( RND (1) * 3) + 2:Y
     =  INT ( RND (1) * 3) + 2:A
     =  INT ( RND (1) * 3) + 2
245 RD =  INT ( RND (1) * 6) + 1:
      ON RD GOTO 250,255,260,265,
     270,275
250 A = 5:P1$ =  LEFT$ (S$(X,A),(
     Y - 1) * 6):L = 29 -  LEN (P
     1$):P2$ =  RIGHT$ (S$(X,A),L
     ):S$(X,A) = P1$ + "O" + P2$:
     GOTO 280
255 A = 1:P1$ =  LEFT$ (S$(X,A),(
     Y - 1) * 6 + 1):L = 29 -  LEN
     (P1$):P2$ =  RIGHT$ (S$(X,A)
     ,L):S$(X,A) = P1$ + "O" + P2
     $: GOTO 280
260 Y = 5:P1$ =  LEFT$ (S$(X,A),(
     Y - 1) * 6 + 3):L = 29 -  LEN
     (P1$):P2$ =  RIGHT$ (S$(X,A)
     ,L):S$(X,A) = P1$ + "O" + P2
     $: GOTO 280
265 Y = 1:P1$ =  LEFT$ (S$(X,A),(
     Y - 1) * 6 + 2):L = 29 -  LEN
     (P1$):P2$ =  RIGHT$ (S$(X,A)
     ,L):S$(X,A) = P1$ + "O" + P2
     $: GOTO 280
270 X = 5:P1$ =  LEFT$ (S$(X,A),(
     Y - 1) * 6 + 4):L = 29 -  LEN
     (P1$):P2$ =  RIGHT$ (S$(X,A)
     ,L):S$(X,A) = P1$ + "O" + P2
     $: GOTO 280
275 X = 1:P1$ =  LEFT$ (S$(X,A),(
     Y - 1) * 6 + 5):L = 29 -  LEN
     (P1$):P2$ =  RIGHT$ (S$(X,A)
     ,L):S$(X,A) = P1$ + "O" + P2
     $: GOTO 280
280 SX = X:SY = Y:SA = A
285  HTAB 10: FLASH
290  VTAB 23: PRINT "HIT ANY KEY
     TO START"
295  HGR : NORMAL
300  IF  PEEK ( - 16384) < 127 THEN
     300
310  POKE  - 16368,00
1000 X =  INT ( RND (1) * 5) + 1:
     Y =  INT ( RND (1) * 5) + 1:
     A =  INT ( RND (1) * 5) + 1:
     FC = 1: GOTO 1220
1010  HOME : VTAB 22: HTAB 18: PRINT
     FC$(FC):A$ = "":D = 0: IF LS
     = 1 THEN  PRINT X,Y,A
1020  VTAB 22: PRINT " TIME :";T:
     FOR TIME = 1 TO 80
1025  IF  PEEK ( - 16384) > 127 THEN
     1030
1027  NEXT :T = T + 1: VTAB 22: PRINT
     " TIME :";T: GOTO 1020
1030  GET A$
```




```
1035  IF A$ = "?" THEN LS = 1
1040  IF A$ = "Q" THEN 1295
1050  IF A$ = "U" THEN D = 1
1060  IF A$ = "D" THEN D = 2
1070  IF A$ = "N" THEN D = 3
1080  IF A$ = "S" THEN D = 4
1090  IF A$ = "E" THEN D = 5
1110  IF A$ = "W" THEN D = 6
1120  IF A$ = "F" THEN 1280
1125  IF A$ = "*" THEN 1287
1130  IF D = 0 THEN 1010
1135 T = T + 1
1140  IF  MID$ (S$(X,A),(Y - 1) *
     6 + D,1) = "X" THEN  PRINT  CHR$
     (7): GOTO 1010
1150  ON D GOTO 1160,1170,1180,11
     90,1200,1210
1160 A = A + 1: GOTO 1220
1170 A = A - 1: GOTO 1220
1180 Y = Y - 1: GOTO 1220
1190 Y = Y + 1: GOTO 1220
1200 X = X + 1: GOTO 1220
1210 X = X - 1: GOTO 1220
1220  IF X > 5 OR X < 1 OR Y > 5 OR
     Y < 1 OR A > 5 OR A < 1 THEN
      PRINT "YOU WIN": GOTO 3000
1230  HCOLOR= 3: HPLOT 0,0: CALL
     62454: HCOLOR= 0: GOSUB 120
1240  FOR I = 1 TO 6: IF  MID$ (S
     $(X,A),(Y - 1) * 6 + I,1) =
     "X" THEN  NEXT : GOTO 1010
1250 R = FC(FC,I) + 1
1260  HCOLOR= 0: ON R GOSUB 125,1
     30,135,140,145,150
1270  NEXT : GOTO 1010
1280  PRINT  TAB( 39); TAB( 0);"W
     HAT FACING (N/S/E/W) ?";
1281  GET A$: IF A$ = "N" THEN FC
      = 1: GOTO 1220
1282  IF A$ = "S" THEN FC = 2: GOTO
     1220
1283  IF A$ = "E" THEN FC = 3: GOTO
     1220
1284  IF A$ = "W" THEN FC = 4: GOTO
     1220
1285  GOTO 1281
1287  PRINT  TAB( 37);
1290  INVERSE : HTAB 18: PRINT SX
     ;" ";SY;" ";SA" ": NORMAL : GOTO
     1020
1295  HOME : TEXT
1300  PRINT "DO YOU WANT TO SAVE
     THIS MAZE": INPUT Y$: IF  LEFT$
     (Y$,1) <  > "Y" THEN  GOTO 3
     030
1310  INPUT "WHAT DO YOU WANT TO
     CALL IT ";N$
1320 D$ =  CHR$ (4)
1325  PRINT D$"OPEN OLD MAZE/";N$
     : PRINT D$"CLOSE OLD MAZE/";
     N$
1330  PRINT D$;"DELETE OLD MAZE/"
     ;N$: PRINT D$;"OPEN OLD MAZE
     /";N$: PRINT D$;"WRITE OLD M
     AZE/";N$
1340  FOR A1 = 1 TO 5: FOR X1 = 1
      TO 5: PRINT S$(X1,A1): NEXT
     : NEXT : PRINT SX: PRINT SY:
      PRINT SA: PRINT T: PRINT FC
1350  PRINT D$;"CLOSE OLD MAZE/";
     N$: GOTO 3030
1360  INPUT "WHAT IS ITS NAME ";N
     $
1365  GOSUB 1500
1370 D$ =  CHR$ (4)
1380  PRINT D$;"OPEN OLD MAZE/";N
     $
1385  PRINT D$"READ OLD MAZE/";N$
1390  FOR A1 = 1 TO 5: FOR X1 = 1
      TO 5: INPUT S$(X1,A1): NEXT
     : NEXT : INPUT SX: INPUT SY:
      INPUT SA: INPUT T: INPUT FC
1400  PRINT D$;"CLOSE OLD MAZE/";
     N$: GOTO 285
1500  HOME : VTAB 10: HTAB 15
1510  INVERSE : PRINT "PLEASE WAI
     T": RETURN
2000  VTAB 12: HTAB 18: INVERSE :
      PRINT "MAZE": NORMAL : VTAB
     22: INPUT "DO YOU WANT INSTR
     UCTIONS ";Y$: IF  LEFT$ (Y$,
     1) <  > "Y" THEN  RETURN
2010  HOME : PRINT "THE OBJECT OF
      MAZE IS TO FIND YOUR WAY": PRINT
     : PRINT "OUT OF A 5X5X5 CUBI
     C MAZE. IN ONE OF THE": PRINT
     "ROOMS THERE IS AN EXIT OUT
     OF THE MAZE."
2020  PRINT : PRINT "YOU MUST TRY
      TO FIND IT IN AS FEW TURNS
     ": PRINT "AS POSSIBLE. THE C
     OMMANDS ARE  :"
2030  PRINT : HTAB 6: INVERSE : PRINT
     "U";: NORMAL : PRINT "-UP";:
      HTAB 17: INVERSE : PRINT "S
     ";: NORMAL : PRINT ":SOUTH"
2040  PRINT : HTAB 6: INVERSE : PRINT
```

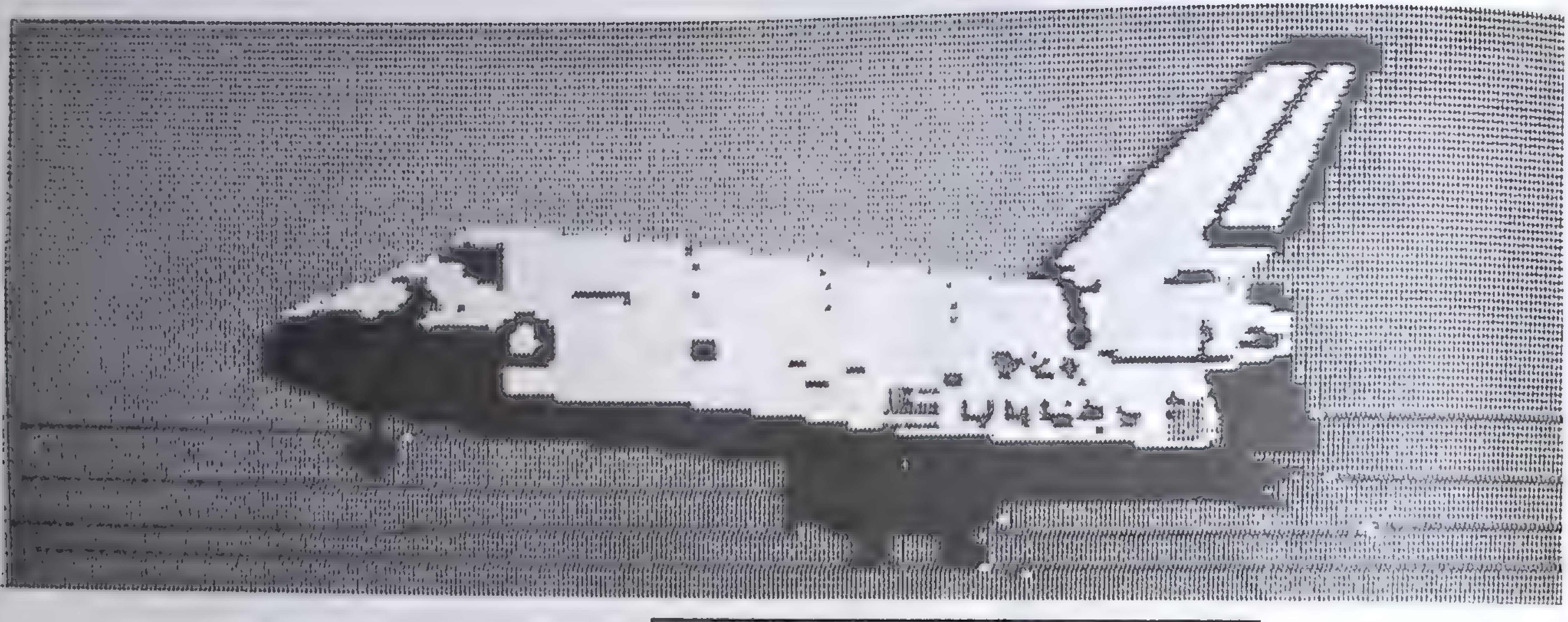


```
"D";: NORMAL : PRINT "-DOWN"
;: HTAB 17: INVERSE : PRINT
"E";: NORMAL : PRINT "-EAST"

2050  PRINT : HTAB 6: INVERSE : PRINT
"N";: NORMAL : PRINT "-NORTH
";: HTAB 17: INVERSE : PRINT
"W";: NORMAL : PRINT "-WEST"

2060  PRINT : HTAB 6: INVERSE : PRINT
"Q";: NORMAL : PRINT "-QUIT"
;: HTAB 17: INVERSE : PRINT
"F";: NORMAL : PRINT "-CHANG
E FACING"
2063  PRINT : HTAB 6: INVERSE : PRINT
"*";: NORMAL : PRINT "-EXIT
POSITION"
2065  PRINT : HTAB 6: INVERSE : PRINT
"?";: NORMAL : PRINT "-DISPL
AY YOUR POSITION"
2070  VTAB 23: PRINT "HIT ";: INVERSE
: PRINT "SPACE";: NORMAL : PRINT
" FOR MORE"
2080  IF  PEEK ( - 16384) < 127 THEN
2080
2090  POKE  - 16368,0: HOME : INVERSE
: PRINT "F";: NORMAL : PRINT
" WILL COME BACK WITH A QUES
TION AS TO": PRINT : PRINT "
WHICH FACING YOU WISH.HIT ON
LY ONE KEY": PRINT : PRINT "
AND ";: INVERSE : PRINT "RET
URN": NORMAL
2100  PRINT : PRINT "PLEASE WAIT
WHILE IT SETS UP THE MAZE": PRINT
: PRINT : RETURN
2632 .17727 INVERSE : HTAB 18: PRINT
X;" ";Y;" ";A: NORMAL
3000  TEXT : HOME : VTAB 5: FLASH
: HTAB 12: PRINT "CONGRATULA
TIONS !"
3005  NORMAL
3010  PRINT : PRINT  TAB( 7)"YOU
HAVE FINISHED THE MAZE IN ":
PRINT  TAB( 7)T;" SECONDS"
3030  INPUT "DO YOU WANT TO PLAY
AGAIN? ";Y$
3040  IF  LEFT$ (Y$,1) = "Y" THEN
RUN
9999  NORMAL
10000  DATA  1,2,4,0,5,3,1,2,0,4,
3,5,1,2,3,5,4,0,1,2,5,3,0,4
```

穿梭機哥倫比亞號機械手特別任務



太空穿梭機「哥倫比亞」號今次的行動是在太空中拾取四散飄浮的太空隕石，以免影响其他太空船的航行安全。由於隕石數量相當多，在行動未完成前，必須留意燃料是否足夠。不過，不用担心，只要遇上一些藍色的太空後勤基地裡，可以進行燃料補充。

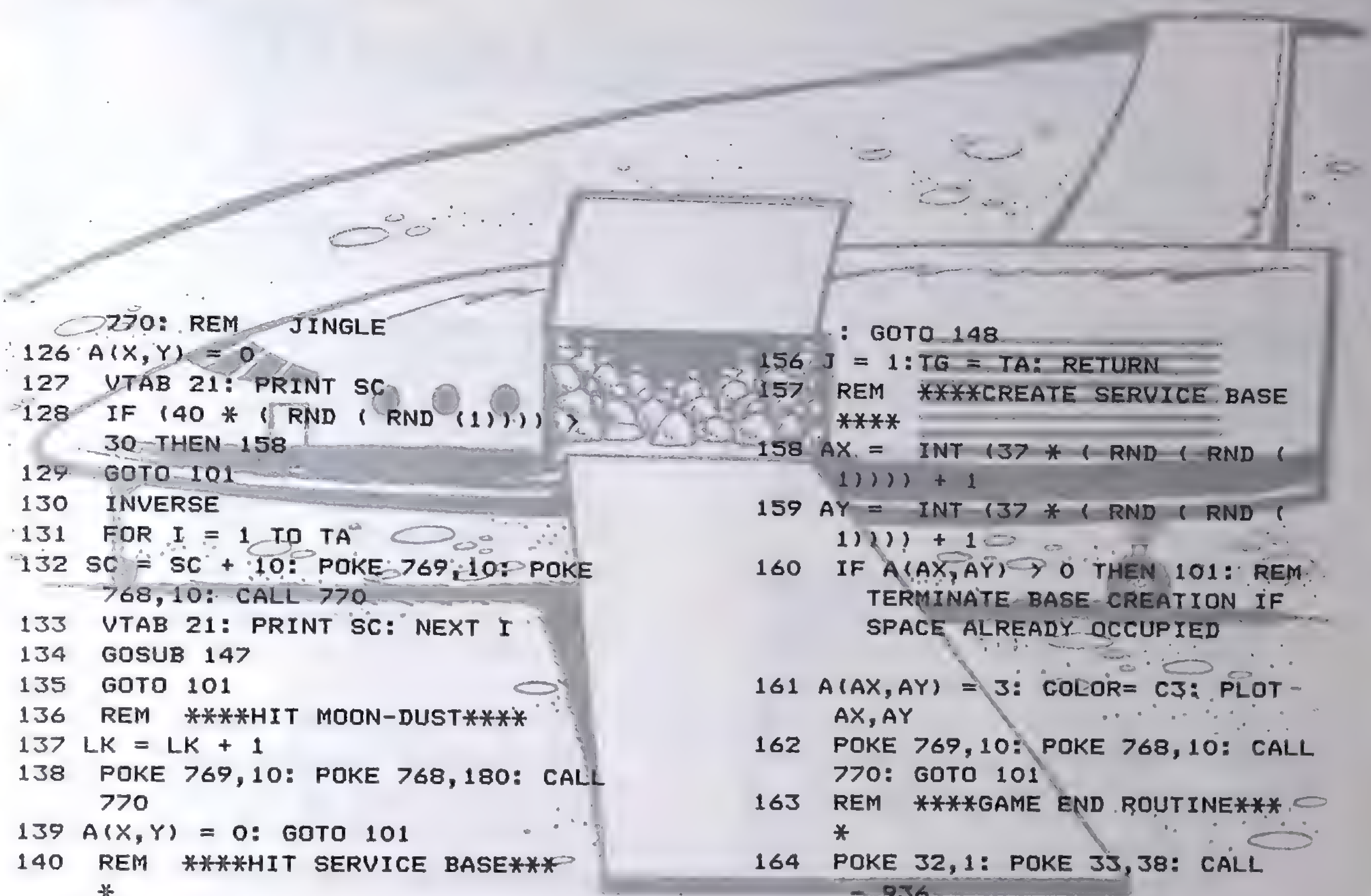
哥倫比亞號移動的方向是由下列按鍵控制：

Z—上　　ESC— 倒後
X—下　　SPACE BAR— 開始

注意：假如閣下使用黑白監視器玩此遊戲，可更改行句183，將變數改爲：

C1=11，C3=9，C4=5，C5=15

```
100  GOTO 182
101  COLOR= C2: PLOT X,Y:X = X +
     1
102  IF  PEEK (M) = 218 THEN Y =
     Y - 1: REM  'UP'
103  IF  PEEK (M) = 216 THEN Y =
     Y + 1: REM  'DOWN'
104  IF  PEEK (M) = 155 THEN 106:
      REM   'HYP'
105  GOTO 111
106  POKE 769,4: POKE 768,200: CALL
     770: POKE 769,4: POKE 768,10
     0: CALL 770
107 FL = FL - (LK + 40)
108  INVERSE : VTAB 23: PRINT FL;
     :: NORMAL : PRINT " "
109 X =  INT (37 * ( RND ( RND (1
     )))) + 1:Y =  INT (37 * ( RND
     ( RND (1)))) + 1
110  GOTO 114
111  IF X > 38 THEN X = 1
112  IF Y < 1 THEN Y = 38
113  IF Y > 38 THEN Y = 1
114  COLOR= C1: PLOT X,Y
115 FL = FL - LK: IF FL < 1 THEN
     164
116  IF FL < 1000 THEN  INVERSE
117  VTAB 23: PRINT FL;: NORMAL :
      PRINT " "
118  IF A(X,Y) = 1 THEN 123: REM
              'STAR-DUST'
119  IF A(X,Y) = 2 THEN 137: REM
             'MOON-DUST'
120  IF A(X,Y) = 3 THEN 141: REM
      'BASE'
121  GOTO 101
122  REM  ****HIT STAR-DUST****
123 SC = SC + 10
124 TG = TG - 1: IF TG = 0 THEN 1
     30: REM  ADD MORE TO BOARD &
     SCORE
125  POKE 769,10: POKE 768,100: CALL
```



```
     770: REM   JINGLE
126 A(X,Y) = 0
127  VTAB 21: PRINT SC
128  IF (40 * ( RND ( RND (1))) >
     30 THEN 158
129  GOTO 101
130  INVERSE
131  FOR I = 1 TO TA
132 SC = SC + 10: POKE 769,10: POKE
     768,10: CALL 770
133  VTAB 21: PRINT SC: NEXT I
134  GOSUB 147
135  GOTO 101
136  REM  ****HIT MOON-DUST****
137 LK = LK + 1
138  POKE 769,10: POKE 768,180: CALL
     770
139 A(X,Y) = 0: GOTO 101
140  REM  ****HIT SERVICE BASE***
     *
141  IF SC - SA < 100 THEN  POKE
     769,10: POKE 768,100: CALL 7
     70: POKE 769,10: POKE 768,20
     0: CALL 770: GOTO 145
142  INVERSE : FOR I = 1 TO  INT
     (24 * ( RND ( RND (1)))) + 1

143 SC = SC + 20:FL = FL + 20: VTAB
     21: PRINT SC: VTAB 23: PRINT
     FL
144  POKE 769,8: POKE 768,255 - (
     I * 10): CALL 770: POKE 769,
     8: POKE 768,255 - (I * 5): CALL
     770: NEXT I
145 A(X,Y) = 0: NORMAL : VTAB 21:
     PRINT SC: VTAB 23: PRINT FL
     :SA = SC: GOTO 101
146  REM  ****SET UP BOARD****
147 TG = TG + 5:TA = TG: COLOR= C
     5
148  FOR I = 1 TO TG
149  POKE 769,10: POKE 768,AX: CALL
     770
150 AX =  INT (37 * ( RND ( RND (
     1)))) + 1
151 AY =  INT (37 * ( RND ( RND (
     1)))) + 1
152  IF A(AX,AY) > 0 THEN 150
153 A(AX,AY) = J: PLOT AX,AY: NEXT
     I
154  IF J = 2 THEN 156
155 J = 2:TG = TG + 10: COLOR= C4
     : GOTO 148
156 J = 1:TG = TA: RETURN
157  REM  ****CREATE SERVICE BASE
     ****
158 AX =  INT (37 * ( RND ( RND (
     1)))) + 1
159 AY =  INT (37 * ( RND ( RND (
     1)))) + 1
160  IF A(AX,AY) > 0 THEN 101: REM
      TERMINATE BASE CREATION IF
      SPACE ALREADY OCCUPIED

161 A(AX,AY) = 3: COLOR= C3: PLOT
     AX,AY
162  POKE 769,10: POKE 768,10: CALL
     770: GOTO 101
163  REM  ****GAME END ROUTINE***
     *
164  POKE 32,1: POKE 33,38: CALL
      - 936
165  FOR I = 0 TO 40
166 AA = AA *  - 1: IF AA =  - 1 THEN
      INVERSE
167  IF AA = 1 THEN  NORMAL
168  VTAB 21: PRINT "            RE
     SERVOIR EMPTY              ": NEXT
     I:AA = 1
169  IF SC <  = HI THEN 171
170  FLASH : VTAB 22: PRINT  SPC(
     38): PRINT : VTAB 22: PRINT
     "   NEW HI-SCORE => ";SC;" U
     NITS":HI = SC: INVERSE : GOTO
     172
171  VTAB 22: PRINT  SPC( 38): PRINT
     : VTAB 22: PRINT "   YOUR SC
     ORE IS ";SC;" UNITS"
172  VTAB 23: PRINT "   DO YOU WA
     NT TO PLAY AGAIN Y/N ?     "
173  VTAB 1: HTAB 38: GET A$
174  IF  ASC (A$) = 78 THEN  TEXT
     : NORMAL : HOME : END
175  IF  ASC (A$) <  > 89 THEN 17
     3
176  CALL  - 936: VTAB 22: PRINT
     "  PLEASE WAIT WHILE BOARD I
     S RESET    "
177  COLOR= C2
178  FOR I = 1 TO 38
179  FOR II = 1 TO 38
180 A(II,I) = 0: PLOT II,I: NEXT
     II: NEXT I
181  GOTO 191
```



```
182 DIM A(39,39)
183 M =  - 16384:C1 = 11:C3 = 7:C
    4 = 4:C5 = 5:AA = 1:J = 1
184  SPEED= 255
185  FOR I = 770 TO 792: READ A: POKE
     I,A: NEXT I: REM  POKE SOUND
      IN
186  TEXT : CALL  - 936
187  INVERSE : VTAB 10: HTAB 10: PRINT
     " S T A R  -  D U S T."
188  PRINT : PRINT "        PRESS S
     PACE BAR TO START       ": WAIT
     49152,32: CALL  - 936: GOTO
     199
189  GR : COLOR= C1
190  HLIN 0,39 AT 0: HLIN 0,39 AT
     39: VLIN 1,39 AT 0: VLIN 1,3
     9 AT 39
191  POKE 32,1: POKE 33,38
192  VTAB 21: PRINT "SCORE  => ";
     : PRINT  SPC( 26): PRINT  SPC(
     40): PRINT "FUEL   => ";: PRINT
      SPC( 28);
193  VTAB 21: HTAB 28: PRINT "STA
     RDUST"
194 FL = 10000:LK = 5:TG = 5:TA =
     0:X = 20:Y = 1:SC = 0:SA = 0

195  POKE 34,20: POKE 33,15: POKE
     32,12: CALL  - 936
196  GOSUB 147: COLOR= C1: PLOT X
     ,Y
197  POKE  - 16368,0: FLASH : PRINT
     : PRINT "  HIT ANY KEY ";: GET
     A$: NORMAL : CALL  - 936
198  GOTO 101
199  PRINT  SPC( 40): PRINT "
           I N S T R U C T I O N S
              ";: PRINT  SPC( 40):
      NORMAL
200  POKE 34,3
202  PRINT " HOWEVER. BEWARE OF C
     OLLIDING WITH ONE  OF THE MA
     NY MOON-DUST FRAGMENTS.
        THESE WILL PUNCTURE THE H
     ULL OF YOUR    SHIP AND CAUS
     E YOUR FUEL TO LEAK AWAY   F
     ASTER WITH EACH COLLISION.":
      PRINT
203  PRINT " EVERY NOW AND THEN,
     A SPECIAL BASE WILLAPPEAR WH
     ICH WILL SUPPLY YOU WITH MOR
     E FUEL AND ALSO AWARD YOU EX
     TER POINTS, IF YOU HAVE GAIN
     ED AT LEAST 100 UNITS OF  ST
     AR-DUST SINCE YOUR LAST DOCK
     ING.": PRINT
204  GOSUB 212
205  PRINT : PRINT " THE MISSION
     BECOMES MORE DIFFICULT AS  Y
     OU WEND YOUR WAY FURTHER. SO
      IF YOU ARETRAPPED AND COLLI
     SION IS IMMINENT. THEN DON'T
      PANIC - YOU CAN ESCAPE INTO
      HYPER-SPACE."
206  PRINT "THIS DOES USE MORE FU
     EL HOWEVER. SO YOU MUST REME
     MBER TO WATCH YOUR GAUGE, AL
     SO IT MUST BE STRESSED THAT
     YOU WILL REMAININ HYPER-SPAC
     E UNTIL YOU RETAKE COMMAND."

207  GOSUB 212
208  INVERSE : PRINT "OBJECT
          COLOR    VALUE
     ": NORMAL : PRINT : PRINT "S
     TART-DUST     => PINK       10
       UNITS": PRINT : PRINT "MOO
     N-DUST     => GREEN       1  F
     UEL UNIT": PRINT
209  PRINT "SERVICE BASE => BLUE
          480 FUEL UNITS
                          480 UNI
     TS"
210  INVERSE : PRINT "CONTROLS :"
     : NORMAL : PRINT : PRINT "
      UP    => 'Z'": PRINT : PRINT
     "    DOWN => 'X'": PRINT : PRINT
     "    HYP  => 'ESC'": PRINT : PRINT
     "    STAR => 'SPACE'=
211  GOSUB 212: TEXT : CALL  - 93
     6: INVERSE : GOTO 189
212  INVERSE : VTAB 23: PRINT "
     PRESS SPACE BAR TO CONTINUE
          "
213  POKE  - 16368,0
214  VTAB 23: HTAB 32: GET A$
215  IF  ASC (A$) <  > 32 THEN 21
     4
216  NORMAL : CALL  - 936: RETURN

217  DATA     173,48,192,136,208,
     5,206,1,3,240,9,202,208,245,
     174,0,3,76,2,3,96,0,0
```

COMPUTER CARTOON MAKING

要有生動的畫面，最簡單的方法是以連貫性的動作來表達圖形。通常模擬活動圖形是一門相當深奧的學問，縱使是各種專業性的程式，也不是可輕易解決。本文是講述活動圖形的設計原理，對於部份初學者來說，可能非常難懂。筆者希望用深入淺出的手法講述究竟如何產生活動圖形的。

利用向量圖形（VECTOR SHAPE）讀者可以製造出一連串任意安排的程序，圖表（數量不拘），甚至可指定某一圖形出現次數的多寡，舉一個簡單的例子：如想製造一個人的行路動作，連串的圖形數目應是4／6／6／8／1／9／3／5／7／，而這十個連串圖形數目，基本上是重複使用的。

圖形表的指數數目，應參照 APPLESOFT 的圖形數目，（無須理會 ASSEMBLY中的圖形數目，而在 BAS—IC 指令的 DRAW 或 X DRAW，圖形是唯一最方便的方法。也許，如果圖形表數目與圖形次序數目相同，這會更爲方便，但若特定的圖形比一個次序的數目爲多時，這點就不太重要了。

以步行爲例，如由側面觀看人的步行動作，可以看到這是幾個重複動作的組合。因此，卡通片也是用幾個重複的圖形製成。利用以下的語句，便可以製造出步行的動作：

（1）ERASE（清除畫面），利用 XDRAW，將圖形繪於舊座標上（OX，OY）。

（2）DRAW（繪畫），利用 XDRAW，圖形會出現在新位置（X，Y）。

（3）將新座標轉貯於舊座標內（OX＝X：OY＝Y）。

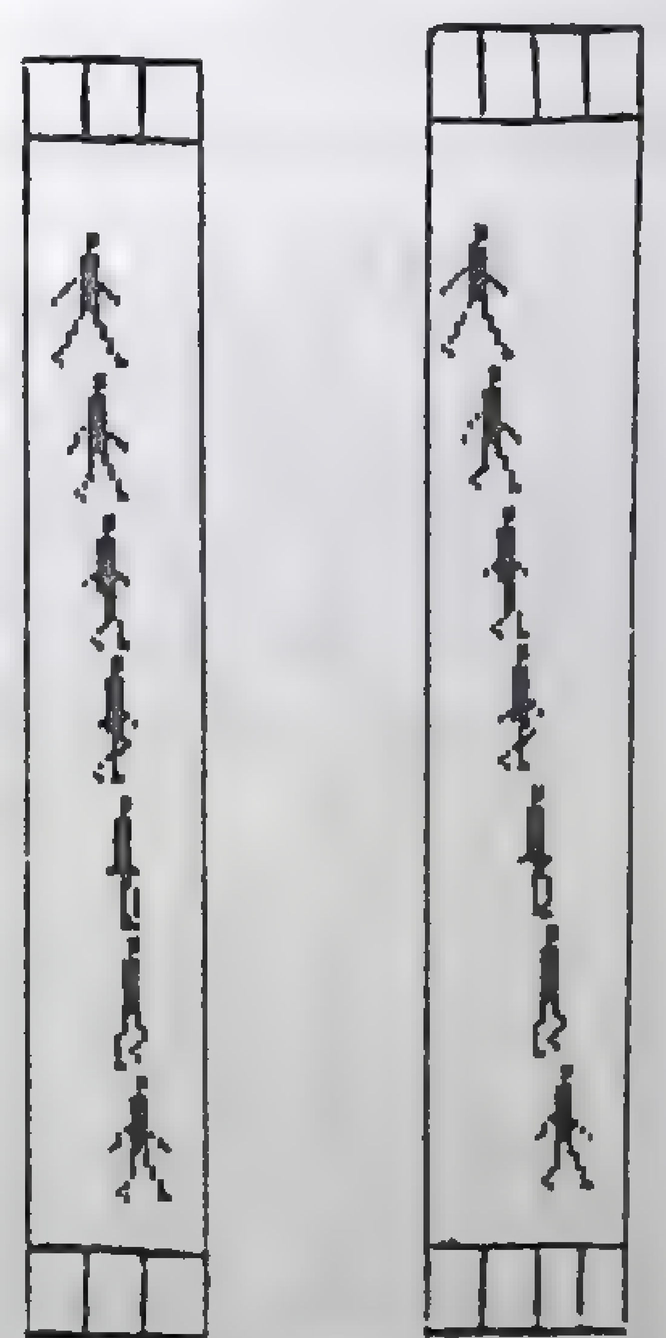
（4）利用 X＝X＋STEP 計算出新的座標，如果步行的水平及垂直動作的距離都是一樣（如上梯級動作），Y就計算成爲Y＝Y＋STEP。

（5）返回第一步

記着，如果讀者是使用跳動方式就將更爲複雜，因爲需要在顯示下一頁的同時在螢幕上繪畫。頁跳動的方法是專針對顯示繪畫工作的進行及開始顯現出繪畫後的結果，其效果是要令到繪畫時看來更自然，不會出現閃動的情形。

到現在應是討論方塊圖形次序的時候了，利用方塊圖形，不單只是在適當的時間，位置作恒値增加步驟（STEP）座標的連串圖形繪畫，且可隨時在螢幕上的任何位置繪上的向量圖形，但其位置必需符合X＜0，X＞279，Y＜0及Y＞191的法則，但方塊圖形（BLOCK SHARE）則不能使用這裡所敍述的方法。

如果是方塊圖形，就必定要保留Y＝0，Y＝191，



圖（1）印出連環活動圖形。

Which Screen, Which Shape, Draw and Erase Chart

screen on which to erase/draw	screen displayed	shape # erased	shape # drawn	HL of shape erased	HL of shape drawn
1	2	1	3	0	0
2	1	2	4	0	0
1	2	3	5	0	0
2	1	4	6	0	0
1	2	5	7	0	0
2	1	6	1	0	1
1	2	7	2	0	1
2	1	1	3	1	1
1	2	2	4	1	1
2	1	3	5	1	1
1	2	4	6	1	1
2	1	5	7	1	1
1	2	6	1	1	2
2	1	7	2	1	2
1	2	1	3	2	2
↓	↓	↓	↓	↓	↓
2	1	5	7	34	34
1	2	6	1	34	0
2	1	7	2	34	0
1	2	1	3	0	0

圖表WHICH SCREEN，WHICH SHAPE，DRAW及ERASE CHART

X BYTE行（COLUMN）（水平位移：HORIZON—TAL OFFSET）0 及 39 。除非你要使用雙對的緩慢移動動作，否則你不能移動少於 1 BYTE 的水平，更詳細資料，可參閱 HI—RES SECRETS 。因此你必需使用次序爲 7 的 PRE—SHIFTED SHAPES（預先移動圖形）。

預先移動圖形次序是一可容許水平移動少於 7 點的方塊圖形次序；如果是相似的圖形物體如 7 個獨立的飛碟，在首個飛碟上，預先移動圖形就很容易行走（RUN）一個自動次序營造程式（列表 1）及將 7 個飛碟次序作爲一個表（TABLE）的貯起來，參看圖 1 ，2 a 及 2 b 。

在圖 1 我們可以看見 STEP 1（先前移動），7 —圖形方塊圖形次序都是 3 BYTE 濶；STEP 2 ，7 —圖形方塊圖形次序都是 4 BYTE 濶，考慮一下這個圖形方塊的左右邊界當作爲眞正的方塊邊緣，注意圖形次序的整個運行（RUN）過程，不論是 X 座標都沒有 IOTA（IOTA：INFORMATION OVERLOAD TESTION A—PPARATUS ：信息過載試驗裝置，而只於次序完畢後將 STEP 數値加入水平 BYTE—COLUMN 座標，參看圖 2 b 。在圖 2 a 我們可以看列 X 座標的遞增是每一次序是 3 。在 2 個圖案，HR 均代表在水平座標， HL代表左水平座標，VT 代表頂垂直座標，VB 代表底直座座標：

```
        VT
     *******
HL   *******  HR
     *******
     *******
        VB
```

再者，由於方塊圖形在水平方面是使用 BYTE—CO—LUME 座標而不是規律化的七座標，所以方塊圖形在螢幕上只可能有 40 個 X 座標，而不是像向量圖形般有 280 個 。

方塊圖形次序

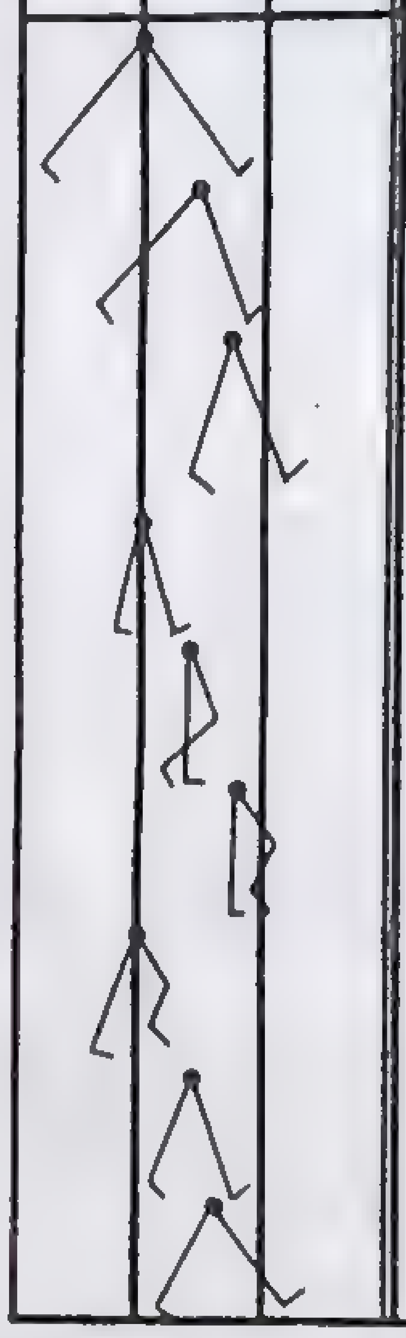
在圖 1 裡，1 至10的圖形是向量圖形表（MAN）裡抽取出來以營造出 MANA 方塊圖形次序表內的 9 個圖形，而這個表是以一種非標準方法在每一次序裡作 3 次的更新數據（UPDATE），但在 MANA 營造 MANC 時，方塊圖形（圖 20 a ）則永遠是作標準 7 —圖形遞增次序更新的。這種步驟都是將不同的向量及方塊圖形載入 SCANA（參看 APPLE COLUMN）內及將其貯在預先計算座標裡（將其貯存並作爲不同圖形表的數字）。

再參照圖 2 a ，注意方塊圖形是 5 BYTE 濶的，但其實與營造 4 BYTE 濶的方法一樣的容易（那個空白 BYTE 是用作爲實試的）。現在，參看第一及第七個圖形，如果我們繼續次序的話，下一個圖形（卽第八個）應該放於那裡及究竟會變成怎麽樣子呢？

首先，我們注意到每一個圖形是將前一個圖形的兩點移向左，隨即會尋找如次序數字 1 的圖形以繼續移動，由於 X ＝ 49 是下一個圖形中心點的水平座標，因此次序內的第一個圖形會正確地在綫的中心點完結，就如圖位置一式一樣。由於方塊圖形（BLOCK SHAPE）的圖象（FIG—URE）會準確地移動兩個 BYTE ，與及每一圖形次序的圖象之間弱點之 STEP 數値是 2 ，這代表了我們將要使用的 STEP 數値相等於所需的水平 BYTE — 座標增値剛在開始次序完畢之前。

MANA

3-byte wide block shape

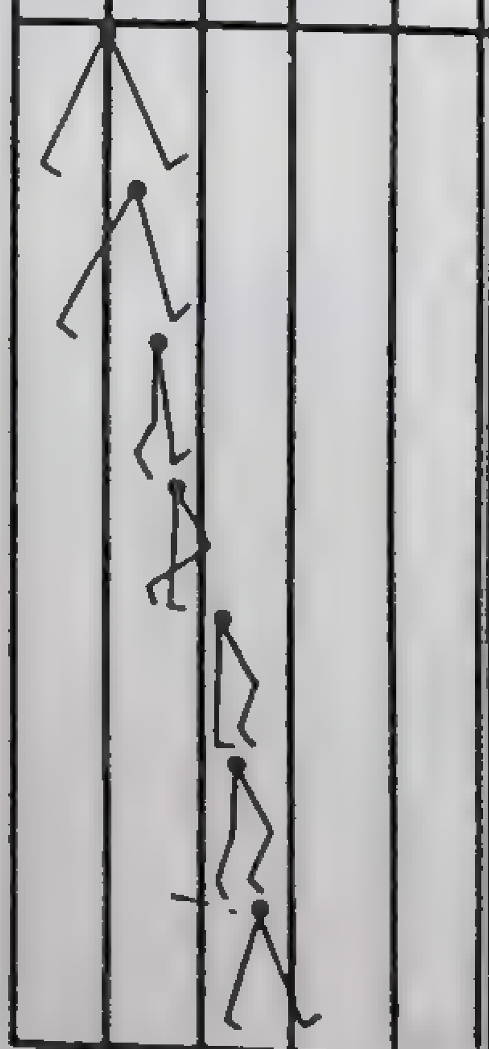


1st byte	2nd byte	3rd byte	sequence #	shape # from MAN	centered on this hor coord	VT,VB	move HL and HR up by 1?	hor coords of block-shape boundaries	MANC seq #
			1	1	35	19,40	yes	28-49	–
			2	6	38	19,40	no	28-49	1
			3	2	40	19,40	no	28 49	2+7
			4	7	42	19,40	yes	35-56	3
			5	3	45	19,40	no	35 56	4
			6	8	47	19,40	no	35 56	5
			7	4	49	19,40	yes	42-63	6
			8	9	52	19,40	no	42-63	-
			9	10	54	19,40	no	42 63	-

圖（2 A）MANA。

MANC

5-byte wide block shape



1st byte	2nd byte	3rd byte	4th byte	5th byte	seq #	shape #	shape centered on hor coord	VT,VB	move HL and HR up by 2?	hor coords of block-shape boundaries
					1	6	35	0,21	yes	28-63
					2	2	37	0,21	no	28 63
					3	7	39	0,21	no	28-63
					4	3	41	0,21	no	28 63
					5	8	43	0 21	no	28 63
					6	4	45	0 21	no	28-63
					7	2	47	0 21	no	28 63

圖（2 B）MANC。

意思是說，在7個圖形顯現之前，所有方塊圖形座標都是一樣的，但只限於次序再開始之前水平 BYTE 一座標才是作2的遞增。究竟會有甚麼發生呢？在效果上，是不是所有方塊圖形內的圖象移動都是局限在方塊圖形的界限內，而不是由座標所處理呢？附帶說明， MANA 及MANC 的所有圖形數字是由取自一一一個行路動作的向量圖形表。當在營造時讀者所參照的形狀數字是使用如 MANC 的方塊圖形表裡1一7的圖形，而相等於1一7的次序數字。

兩頁的跳動

當讀者使用不同的方塊圖形次序的預先移動圖形及利用兩頁 跳動以營造較完美的畫面時是比較困難許多的。細察一下表 （TABLE），你就會發現很多地方都是很笨拙的，如繪畫一個畫面時需要顯示另一個畫面，而一個畫面的次序是1，3，5，7，2，4，6，1，3，而另一個畫面次序則會 是2，4，6，1，3，5，7等等。

最特別的地方是抹洗圖形6時，當< SINGLE A-STERISK > HL < 水平左 BYTE 座>時是0。第一，當繪畫圖形1時，必需返回0以抹洗另一個畫面的圖形7（DOUBLE ASTERISKS），另外是增至1以繪圖形2。在編寫這些動作程式時，最重要是將所發生的（如圖形的移動）製成易於追查的圖表。

現在，如果讀者輸入列表4及5的 TEST F（CA-LL 36934） TEST G （ CALL 36934） 及列表3與2的MANC 及 ASMINPUT ，將可獲得利用7一圖形方塊次序作爲主左右移動的20 頁跳動 方塊圖形次序應用程式。以下是一些 BSAVE 地址及長度給予在這些列表中的不同檔案：

列表3　MANC，A$900，L1646（ STEP 2，21高，4濶，7圖形）。

列表4　TEST F（CALL 36934），A36864，L 324 。

列表5　TEST G （ CALL 36934），A36864，L 342 。

列表6　TEST H（ CALL 2186），A 2048，L 224 。

當輸入 MANC 時，可以省略了如 $970 至$9FF的數據，而只輸入所提供的數據。

在運行程式前，必須先按入 POKE 103，1：POKE 104，96：POKE 24576，0。

現在讓我們利用MANC 的7個圖形的，TEST F（CALL 36934）及 TEST G （CALL 36934）的程式來營造出人行路連串活動動作（這些程式的效果是右左移動的動作）。所用的驅動程式（現在所運行的）是 ASMI-NPUT，將圖形表命爲 MANC 。（相信讀者已保留了所需的檔案了），比方說 RIGHT WARD 是所行走的方向，其指定濶度是4，高度是 21 ，STEP 是2，右邊界限是34 ，第一個圖形是次序1，再給予一個延遲循環，其高 BYTE 是 70 ，另一個延遲循環的低 BYTE 是 255。

這個程式的結果是正常步行速度，且沒有閃動的情形出現（因爲這個程式是使用兩頁跳動法來模擬動作的），但邪個兩步移動時動作則很礙眼，改善方法是將次序內使用 14 個圖形及 1 STEP （步驟）移動，在次序的結尾仍然保留着使用2爲水平 BYTE 座標的遞增值，但却減低一半的延遲時間。注意列當 HR（水平右座標）達到 34 時，次序即再次重新開始。34 的值是從上述　爲的地方輸入的，只需按下任何的鍵即會再次工作，將延遲循環的高 BYTE 値改爲35 以使其移動速度加快一倍（可省略了 $EE：34 及其他的螢幕上的數據，這些數據是用來指出當動作停止時變值是如何的工作）。

現在可以見到人像在螢幕上步行了，且發現方塊圖形參數內的 STEP 移動已不復存在了。現在嘗試使用 STEP 値爲1及延遲循環的高 BYTE 是 70 的效果，注意 ST-EP 數值1與2所做成圖形的分別；同時留意方塊內 ST-EP 的移動，每一次是一個點的，並沒有 STEP 値是1的那麼明顯。在隨後的兩個實驗中，是用以證明如果使用 14 個一步圖形及水平 BYTE 座標遞增值是2時，所得到的效果是最完美的。（附帶地說明，當向左移動時，選擇 LE-FTWARD，但其他輸入時，應選擇如 RIGHTWARD）的一樣。）

次序的營造

現在嘗試輸入列表6及1的 TEST H（CALL 2168）及次序營造程式。1（切勿忘記先前曾提及過的POKE 104，96 等等），跟着是按回車以運行次序營造程式，選擇（1）以載入方塊圖形表，並將圖形表命名爲 MANC，指定圖形數目是1，VTOP 是 10，VBOT 是 31，HRIGHT是5，HLEFT是1，當被電腦詢問是否需要更多的圖形時，即輸入 NO。

在MENU 裡，選擇（3） DEFINE BLOCK S-HAPE WITH PADDLE （利用遊戲棒決定方塊圖形），細讀索引後利用遊戲棒等找出那一個是令列點浮標作水平的移動，我們稱這枝遊戲棒爲X，而另一枝則爲Y；移動點浮標直至剛離開人形圖形週圍假設的一個四方形方塊的左上角，按下遊戲棒的按鈕；移向至右下角並按下Y遊戲棒的按鈕一一但不是直至到已經移動了7個點到該位置才停止，因爲需要預留空位給方塊內的 STEP移動。（如果是使用14一點位移（OFFSET），STEP 値應是2，倘若位移是 21 一點，則使用3的 STEP 値），當方塊圖形的STEP 値已作7次向右（下）變動後即按Y的按鈕。

當被詢問正方形（用以決定方塊圖形的參數）是否妥當（OKAY），答是（YES）或否（ NO），答覆後一切已成定局，不能再答。現在可選擇（2）以決定方塊圖形次序及保留所有次序的水平 STEP 値，這裡應選擇的 STEP 値，這裡應選擇的STEP 値是1。圖形次序時數量是7，次序將被保留的第一個方塊圖形的數目是1，當被問及“是否已準備好這個次序？”時，答Y（ YES），隨即可以見到7個圖形利用移動（ SHIFT）而被做成，（在此之後，

相關的圖形會被審視，所得到的數據將有貯入記憶內）。以TEST 作為檔案名稱，7 用作圖形表內最後的圖形。

當次次序被貯存好後，可出運行 ASMINPUT 時使用，以用作檢査稍後的輸出。除非在次序造時使用了較大的數目，否則 STEP 值一定要使用 1。如沒有任何錯誤時，那個卡通人的動作應該是很圓滑的，當然，如果讀者能利用上述的技巧再創出一些更美麗的圖案，那就最好不過了，亦希望讀者能將程式投函本刊刊登，以供其他讀者分享！

最後，筆者再要強調，本文程式純粹是從原理設計方向出發，對於部份讀者來說可能太深奧。□

程式（1）SEQUENCE CREATOR

```
]LIST

0  ONERR  GOTO 63990
1  PRINT  CHR$ (4);"BLOADTEST H":
     GOSUB 2500: GOTO 600
2  HOME : INPUT "SHAPE TABLE NAME
    : ";STN$: POKE 7,SHN
4 D$ =  CHR$ (4): PRINT D$"BLOAD"
   ;STN$
5  HOME : VTAB 21: INPUT "SHAPE #
    : ";SHN: POKE 7,SHN
15  POKE  - 16304,0: POKE  - 1629
    7,0
18 US = 1:BS = 0
20  INPUT "VTOP:";VT: INPUT "VBOT
    :";VB: INPUT "HRIGHT:";HR: INPUT
    "HLEFT:";HL
30  POKE 252,VT: POKE 253,VB: POKE
    254,HR: POKE 255,HL
42  CALL 2116
43  HOME : VTAB 21: INPUT "DO YOU
     WANT ANOTHER SHAPE? (Y/N):"
    ;QW$: IF  LEN (QW$) = 0 THEN
    43
44  IF  ASC (QW$) <  > 89 THEN  HOME
    : VTAB 21: GOSUB 63000: GOTO
    600
45  GOTO 5
47  POKE  - 16303,0: POKE  - 1629
    8,0: HOME : VTAB 1: PRINT "U
    SE THE PADDLES TO MOVE THE D
    OT TO THE UPPER LEFT RECTANG
    LE POINT. HIT PDL 0 BUTTON.
    THEN MOVE THE DOT TO THE LOW
    ER RIGHT RECTANGLE POINT. HI
    T PDL 1 BUTTON.": GOSUB 6300
    0
48  POKE 232,248: POKE 233,8: SCALE=
    1: ROT= 64
49  POKE  - 16304,0: POKE  - 1629
    7,0
50  HOME :P1 =  PDL (1): IF P1 >
    150 THEN 50
55 P0 =  PDL (0): XDRAW 1 AT P0,P
    1:X% = P0:Y% = P1
60 P1 =  PDL (1): IF P1 > 150 THEN
    60
65  FOR QW = 1 TO 200: NEXT : HOME
    : VTAB 21: PRINT "X: "P0: PRINT
    "Y:"P1
70 P0 =  PDL (0): XDRAW 1 AT X%,Y
    %: XDRAW 1 AT P0,P1:X% = P0:
    Y% = P1
80 B0 =  PEEK ( - 16287): IF B0 >
    127 AND FL = 0 THEN FL = 1: GOTO
    100
85 B1 =  PEEK ( - 16286): IF B1 >
    127 AND SG = 0 THEN SG = 1: GOTO
    110
90  GOTO 60
100 VT = P1:HL =  INT (P0 / 7): PRINT
     CHR$ (7): IF SG = 1 THEN 12
    0
105  GOTO 60
110 VB = P1:HR =  INT (P0 / 7): PRINT
     CHR$ (7): IF FL = 1 THEN 12
    0
115  GOTO 60
120  HOME : VTAB 21: PRINT "HOR.-
    -FROM:"HL" TO "HR"---WIDTH:"
    (HR - HL)
125  XDRAW 1 AT P0,P1
130  PRINT "VER.--FROM:"VT" TO "V
    B"---HEIGHT:"(VB - VT): VTAB
    23: PRINT "JOT THIS DOWN! (H
    IT ANY KEY TO CONT.)": GOSUB
    63010
150  POKE 252,VT: POKE 253,VBOT: POKE
    254,HRIGHT: POKE 255,HLEFT
155  HCOLOR= 3
160  HPLOT 7 * HRIGHT + 7,VT TO 7
     * HRIGHT + 7,VB TO 7 * HLEF
    T,VB TO 7 * HLEFT,VT TO 7 *
    HRIGHT + 7,VT
170  IF ZQ = 1 THEN  RETURN
180  PRINT : INPUT "IS THE RECTAN
    GLE DONE O.K.? (Y/N):";AN$: IF
     LEN (AN$) = 0 THEN 180
185  IF  ASC (AN$) = 78 THEN SG =
    0: HCOLOR= 0:FL = 0:ZQ = 1: GOSUB
    160:ZQ = 0: HCOLOR= 3: GOTO
    50
191  GOTO 600
204  HOME : VTAB 21
205  PRINT "SHAPE # "ST
208  POKE 7,ST: POKE  - 16304,0: POKE
     - 16297,0
210  IF QZ = 0 THEN QZ = 1:ZQ = 1
    : HCOLOR= 0: GOSUB 160: HCOLOR=
    H
215 NN = NN + 1
220  CALL 2048
225  IF NN >  = NS THEN 300
240  FOR QQ = 1 TO SS: CALL 2186:
```

```
    NEXT
245 ST = ST + 1
250  GOTO 204
300 D$ =  CHR$ (4)
301  VTAB 21
302  INPUT "FILE NAME: ";N$: IF  LEN
     (N$) = 0 THEN 302
303  INPUT "DID YOU GET IT RIGHT?
      (Y/N):";Z$: IF  LEN (Z$) =
     0 THEN 302
304  IF  ASC (Z$) <  > 89 THEN 30
     2
307  TEXT : VTAB 1: HOME : GOSUB
     5040
308 LL = 256 * LS
309 PL = LS
310  PRINT D$"BSAVE";N$;",A2304,L
     ";LL
312  PRINT "LAST SHAPE AND ALL TH
     E SHAPES THAT CAME BEFORE IT
      TOOK UP "LL" BYTES.": PRINT
     "LAST SHAPE:";LS: PRINT "(HI
     T ANY KEY TO CONTINUE):": GOSUB
     63010
400  GOTO 600
402  HOME : VTAB 21: INPUT "STEP
     SIZE: ";SS
403  PRINT : INPUT "# OF SHAPE IN
      SEQUENCE: ";NS: PRINT : INPUT
     "# OF 1ST BLOCK-SHAPE IN SEQ
     UENCE TO BE SAVED: ";ST
404  PRINT : INPUT "READY TO BEGI
     N AUTOMRTIC SCAN & SAVE  PRO
     CESS FOR THIS SEQUENCE? (Y/N
     ):";QW$: IF  LEN (QW$) = 0 THEN
     600
405  IF  ASC (QW$) <  > 89 THEN 6
     00
410  GOTO 204
600  POKE  - 16303,0: POKE  - 162
     98,0: HOME : VTAB 1: INVERSE
     : HTAB 18: PRINT "MENU:": NORMAL

601 SG = 0:FL = 0:ZQ = 0:QZ = 0:N
     N = 0
602  SCALE= S: HCOLOR= H: ROT= R
603  PRINT "(HIT ESC TO QUIT)": PRINT

605  PRINT "(0)ABORT SCREEN---STA
     RT OVER": PRINT
610  PRINT "(1)LOAD BLOCK SHAPE T
     ABLE": PRINT
640  PRINT "(2) GIVE HOR. STEP SI
     ZE FOR BLOCK-SHAPE   SEQUENC
     E & SAVE ENTIRE SEQUENCE": PRINT

650  PRINT "(3)DEFINE BLOCK SHAPE
      WITH PADDLES": PRINT
660  PRINT "(4)VIEW SCREEN": PRINT
690  FLASH : PRINT "(CHOOSE 0-4):
     ";: NORMAL : GET A$: PRINT  CHR$
     (13)
692  IF  ASC (A$) = 27 THEN  TEXT
     : HOME : END
700  IF  LEN (A$) = 0 THEN 690
710  IF  VAL (A$) < 0 OR  VAL (A$
     ) > 4 THEN 690
719  IF A$ = "0" THEN 912
720  ON  VAL (A$) GOTO 2,402,47,9
     20,600
721  GOTO 600
912  INPUT "SURE YOU WANT TO ABOR
     T SCREEN? (Y/N):";QW$: IF  LEN
     (QW$) = 0 THEN 912
913  IF  ASC (QW$) <  > 89 THEN 6
     00
914  HGR : GOTO 600
920  POKE  - 16304,0: POKE  - 162
     97,0: VTAB 21: GOSUB 63000: GOTO
     600
2500  POKE 2296,1: POKE 2297,0: POKE
     2298,4: POKE 2299,0: POKE 23
     00,4: POKE 2301,0
2510  POKE  - 16301,0
2511  POKE  - 16303,0: POKE  - 16
     298,0: INVERSE : PRINT "IF Y
     OU ENTERED THIS PROGRAM WITH
      SOME - THING ON THE HI-RES
     SCREEN YOU WANTED TOSAVE, HI
     T THE SPACE BAR NOW---    OT
     HERWISE HIT ANY KEY EXCEPT T
     HE SPACE BAR.": NORMAL
2512 PK =  PEEK ( - 16384): IF PK
      > 127 THEN  POKE  - 16368,0
     : GOTO 2514
2513  GOTO 2512
2514  IF PK = 160 THEN 2520
2515  HGR
2520  RETURN
5040  HOME : VTAB 21: INPUT "# OF
      LAST SHAPE IN BLOCK-SHAPE T
     ABLE: ";LS: IF LS < 1 OR LS >
     23 THEN 5040
5050  RETURN
63000  PRINT "       (HIT ANY KEY
     TO CONTINUE):            "
63010 PK =  PEEK ( - 16384): IF P
     K > 127 THEN  POKE  - 16368,
     0: RETURN
63020  GOTO 63010
63990  PRINT  CHR$ (7): POKE 216,
     0
63991 PP =  PEEK (222): IF PP = 2
     54 THEN  RESUME
63994  POKE  - 16303,0: POKE  - 1
     6298,0
63995  PRINT "YOUR ERR_R IS CODE
     #:"PP: GOSUB 63000: CALL 549
     15: GOTO 600
```

程式（2）ASMINPUT

```
]LIST

0  ONERR  GOTO 63990
1  POKE 8,0: REM  8 MUST BR OED F
     OR THIS PROG. TO WORK!!!!!!!
     !!!!!!!
10 D$ =  CHR$ (4)
25  TEXT : INPUT "SHAPE TABLE NAM
     E: ";N$: IF  LEN (N$) = 0 THEN
     25
27  PRINT : INPUT "YOU WANT YOUR
     SHAPE TO TRAVEL:    (1) -----
     > RIGHT WARDS      (2) <-----
     ---LEFTWARDS        (1-2):";
     Q: IF Q < 1 OR Q > 2 THEN 27

28  IF Q = 1 THEN  PRINT  CHR$ (4
     );"BLOADTEST F"
29  IF Q = 2 THEN  PRINT  CHR$ (4
     );"BLOADTEST G"
30  PRINT : PRINT "INSERT YOUR SH
     APE TABLE DISK NOW:": GET A$
     : PRINT  CHR$ (13): CALL 100
     2: PRINT D$"BLOAD";N$: PRINT
     "ADDRESS: " PEEK (43634) +  PEEK
     (43635) * 256: PRINT "LENGTH
     :" PEEK (43616) +  PEEK (436
     17) * 256
31  INPUT "WIDTH:";WD: INPUT "HEI
     GHT:";HT: INPUT "STEP SIZE:"
     ;SS: INPUT "RIGHT BOUNDARY O
     F LEFT SIDE OF SHAPE:";RB: IF
     Q = 1 THEN  POKE 235,WD - SS
     : POKE 29,(RB - SS) + WD: POKE
     30,RB - SS
32  INPUT "# OF 1ST SHAPE IN SEQU
     ENCE: ";SH: POKE 239,WD: POKE
     238,RB: POKE 237,HT: POKE 23
     6,SS: IF Q = 2 THEN  POKE 25
     ,39 - WD: POKE 235,SS + 39
33  IF Q = 1 THEN  POKE 36955,SH +
     1: POKE 36987,SH: POKE 36994
     ,SH + 1: POKE 37029,SH + 5: POKE
     37046,SH: POKE 37080,SH - 1:
      POKE 37084,SH + 6: POKE 371
     33,SH + 1
34  IF Q = 2 THEN  POKE 37092,SH +
     7: POKE 37096,SH: POKE 37152
     ,SH + 5: POKE 36955,SH + 5: POKE
     36987,SH + 6: POKE 36994,SH +
     5: POKE 37029,SH + 1: POKE 3
     7046,SH + 6
35  TEXT : INPUT "DELAY LOOP HI B
     YTE (1-255):";A: IF A < 1 OR
     A > 255 THEN 35
36  POKE 9,A
37  PRINT : INPUT "DELAY LOOP LO
     BYTE (1-255):";B: IF B < 1 OR
     B > 255 THEN 37
38  POKE 31,B
40  CALL 36934
42  HOME : POKE  - 16368,0
43  TEXT : PRINT "$EF:" PEEK (239
     ): PRINT "$EE:" PEEK (238): PRINT
     "$ED:" PEEK (237): PRINT "$E
     C:" PEEK (236): PRINT "$EB:"
      PEEK (235): PRINT "$1D:" PEEK
     (29): PRINT "$1E:" PEEK (30)
     : PRINT : PRINT
45  TEXT : INPUT "DO YOU WANT TO
     SEE IT SOME MORE? (Y/N):";Q$
     : IF  LEN (Q$) = 0 THEN 45
46  IF  ASC (Q$) <  > 89 THEN  END

50  GOTO 31
63990  POKE 216,0
63991  ONERR  GOTO 63990
63992 PK =  PEEK (222): IF PK = 2
     54 THEN  RESUME
63995  GOTO 0
```

程式（3）MANC

```
*900.96F

0900- 00 00 00 00 00 00 00 00
0908- 30 08 00 00 00 10 04 00
0910- 00 00 08 08 00 00 00 08
0918- 10 00 00 00 04 10 00 00
0920- 00 04 20 00 00 00 02 20
0928- 00 00 00 02 40 00 00 00
0930- 03 40 00 00 00 13 44 00
0938- 00 00 0B 48 00 00 00 07
0940- 50 00 00 00 03 60 00 00
0948- 00 03 40 00 00 00 03 40
0950- 00 00 00 03 40 00 00 00
0958- 01 00 00 00 00 03 00 00
0960- 00 00 03 00 00 00 00 00
0968- 00 00 00 00 00 00 B8 0B
```

```
*A00.A6F

0A00- 00 00 00 00 00 00 00 00
0A08- 00 00 00 00 00 61 00 00
0A10- 00 00 20 40 00 00 00 21
0A18- 00 00 00 00 12 00 00 00
0A20- 00 12 00 00 00 00 0A 00
0A28- 00 00 00 0A 00 00 00 00
0A30- 4E 20 00 00 00 2E 40 00
0A38- 00 00 1E 40 00 00 00 0F
0A40- 00 00 00 00 0E 00 00 00
0A48- 00 0E 00 00 00 00 0E 00
0A50- 00 00 00 0E 00 00 00 00
0A58- 04 00 00 00 00 0C 00 00
0A60- 00 00 0C 00 00 00 00 00
0A68- 00 00 00 00 00 00 00 00
```

```
*B00.B6F

0B00- 00 00 00 00 00 00 00 01
0B08- 48 00 00 00 00 44 00 00
0B10- 00 00 48 00 00 00 00 50
0B18- 00 00 00 00 30 00 00 00
0B20- 00 30 00 00 00 00 30 00
0B28- 00 00 00 30 00 00 00 00
0B30- 38 00 00 00 01 3A 00 00
0B38- 00 00 7C 00 00 00 00 38
0B40- 00 00 00 00 38 00 00 00
0B48- 00 38 00 00 00 00 38 00
0B50- 00 00 00 38 00 00 00 00
0B58- 10 00 00 00 00 30 00 00
0B60- 00 00 30 00 00 00 00 00
0B68- 00 00 00 00 00 00 00 00
```

```
*C00.C6F

0C00- 00 00 00 00 00 00 00 00
0C08- 00 00 00 00 01 50 00 00
0C10- 00 00 00 00 00 00 00 00
0C18- 00 00 00 00 60 00 00 00
0C20- 01 40 00 00 00 02 40 00
0C28- 00 00 01 40 00 00 00 01
0C30- 68 00 00 00 05 70 00 00
0C38- 00 03 60 00 00 00 01 60
0C40- 00 00 00 01 60 00 00 00
0C48- 01 60 00 00 00 01 60 00
0C50- 00 00 01 60 00 00 00 00
0C58- 40 00 00 00 01 40 00 00
0C60- 00 01 40 00 00 00 00 00
0C68- 00 00 00 00 00 00 00 00
```

```
*D00.D6F

0D00- 00 00 00 00 00 00 00 0E
0D08- 00 00 00 00 06 00 00 00
0D10- 00 0A 00 00 00 00 0A 00
0D18- 00 00 00 0A 00 00 00 00
0D20- 0A 00 00 00 00 06 00 00
0D28- 00 00 06 00 00 00 00 0F
0D30- 40 00 00 00 87 00 00 00
0D38- 00 A7 00 00 00 00 07 00
0D40- 00 00 00 07 00 00 00 00
0D48- 07 00 00 00 00 07 00 00
0D50- 00 00 07 00 00 00 00 02
0D58- 00 00 00 00 06 00 00 00
0D60- 00 06 00 00 00 00 00 00
0D68- 00 00 00 00 00 00 00 00
```

```
*E00.E6F

0E00- 00 00 00 00 00 00 00 06
0E08- 00 00 00 00 12 00 00 00
0E10- 00 0A 00 00 00 00 12 00
0E18- 00 00 00 22 00 00 00 00
0E20- 24 00 00 00 00 14 00 00
0E28- 00 00 14 00 00 00 00 1C
0E30- 00 00 00 00 1C 00 00 00
0E38- 00 1C 00 00 00 00 1C 00
0E40- 00 00 00 1C 00 00 00 00
0E48- 1C 00 00 00 00 1C 00 00
0E50- 00 00 1C 00 00 00 00 08
0E58- 00 00 00 00 18 00 00 00
0E60- 00 18 00 00 00 00 00 00
0E68- 00 00 00 00 00 00 00 00
```

```
*F00.F6F

0F00- 00 00 00 00 00 00 00 00
0F08- 00 00 00 06 08 00 00 00
0F10- 02 04 00 00 00 02 08 00
0F18- 00 00 01 10 00 00 00 01
0F20- 10 00 00 00 00 50 00 00
0F28- 00 00 50 00 00 00 04 72
0F30- 00 00 00 02 74 00 00 00
0F38- 01 74 00 00 00 00 78 00
0F40- 00 00 00 07 00 00 00 00
0F48- 70 00 00 00 00 70 00 00
0F50- 00 00 70 00 00 00 00 20
0F58- 00 00 00 00 60 00 00 00
0F60- 00 60 00 00 00 00 00 00
0F68- 00 00 00 00 00 00 0F 90
```

程式（6）TEST H （CALL 2186 ）

```
*800.8E2

0800- A0 09 A6 07 CA E0 00 F0
0808- 04 C8 4C 04 08 98 85 FB
0810- A9 00 85 FA A5 FD 85 06
0818- A2 00 A0 00 20 11 F4 A4
0820- FE A2 00 B1 26 81 FA 88
0828- 18 E6 FA D0 02 E6 FB C0
0830- FF F0 04 C4 FF B0 EC C6
0838- 06 A5 06 C9 FF F0 04 C5
0840- FC B0 D5 60 A0 09 A6 07
0848- CA E0 00 F0 04 C8 4C 48
0850- 08 98 85 FB A9 00 85 FA
0858- A5 FD 85 06 A2 00 A0 00
0860- 20 11 F4 A4 FE A2 00 A1
0868- FA 51 26 91 26 88 18 E6
0870- FA D0 02 E6 FB C0 FF F0
0878- 04 C4 FF B0 EA C6 06 A5
0880- 06 C9 FF F0 04 C5 FC B0
0888- D3 60 A5 FD 85 06 A2 00
0890- A0 00 20 11 F4 18 A4 FF
0898- A9 00 85 08 85 CF B1 26
08A0- 2A 91 26 B0 02 90 02 E6
08A8- 08 C9 80 B0 02 90 02 E6
08B0- CF A5 08 D0 09 B1 26 29
08B8- 7F 91 26 4C C4 08 B1 26
08C0- 09 80 91 26 C4 FE F0 09
08C8- C8 18 A5 CF C9 01 4C 98
08D0- 08 C6 06 A5 06 C9 FF F0
08D8- 04 C5 FC B0 B1 60 00 00
08E0- 3A 22 3B
*
```

程式（4）TEST F （CALL 36934）

```
*
*
]CALL-151

*9000.9144

9000- A0 09 A6 07 CA E0 00 F0
9008- 04 C8 4C 04 90 98 85 FB
9010- A9 00 85 FA A5 FD 85 06
9018- A2 00 A0 00 20 11 F4 A4
9020- FE A2 00 A1 FA 51 26 91
9028- 26 88 18 E6 FA D0 02 E6
9030- FB C0 FF F0 04 C4 FF B0
9038- EA C6 06 A5 06 C9 FF F0
9040- 04 C5 FC B0 D3 60 20 E2
9048- F3 A9 00 8D 52 C0 20 D8
9050- F3 A9 40 85 E6 A9 00 8D
9058- 54 C0 A9 02 85 07 A9 00
9060- 85 FF A5 EF 85 FE A9 00
9068- 85 FC A5 ED 85 FD 20 00
9070- 90 A9 00 8D 55 C0 A9 20
9078- 85 E6 A9 01 85 07 20 00
9080- 90 A9 02 85 07 A9 20 85
9088- E6 A9 00 8D 55 C0 20 A0
9090- 90 A9 40 85 E6 A9 00 8D
9098- 54 C0 20 A0 90 4C 85 90
90A0- C6 07 A5 07 C9 06 D0 2D
90A8- 20 00 90 A4 09 A6 1F CA
90B0- D0 FD 88 D0 F8 A9 01 85
90B8- 07 A4 EC E6 FE E6 FF 88
90C0- D0 F9 18 A5 FF C5 EE 90
90C8- 08 A5 EF 85 FE A9 00 85
90D0- FF 20 00 90 60 A5 07 C9
90D8- 00 D0 48 A9 07 85 07 A4
90E0- EC C6 FE C6 FF 88 D0 F9
90E8- A5 FE C5 EB D0 08 A5 1E
90F0- 85 FF A5 1D 85 FE 20 00
90F8- 90 A4 09 A6 1F CA D0 FD
9100- 88 D0 F8 A4 EC E6 FE E6
9108- FF 88 D0 F9 A9 02 85 07
9110- 18 A5 FF C5 EE 90 08 A5
9118- EF 85 FE A9 00 85 FF 20
9120- 00 90 60 20 00 90 A4 09
9128- A6 1F CA D0 FD 88 D0 F8
9130- E6 07 E6 07 20 00 90 18
9138- AD 00 C0 C9 80 B0 01 60
9140- 68 28 60 00 C6
```

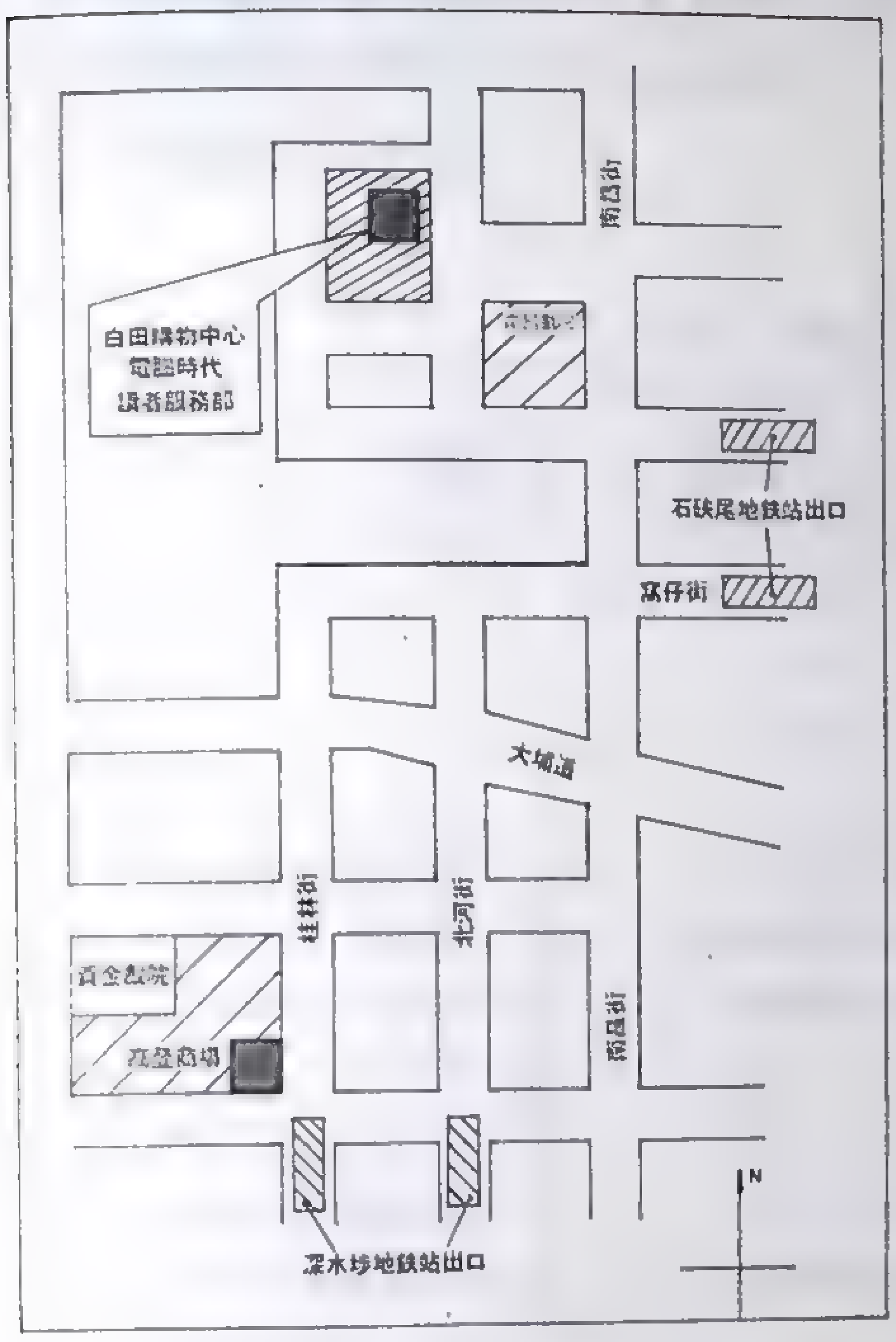
程式（5）TEST G （CALL 36934）

```
]CALL-151

*9000.9157

9000- A0 09 A6 07 CA E0 00 F0
9008- 04 C8 4C 04 90 98 85 FB
9010- A9 00 85 FA A5 FD 85 06
9018- A2 00 A0 00 20 11 F4 A4
9020- FE A2 00 A1 FA 51 26 91
9028- 26 88 18 E6 FA D0 02 E6
9030- FB C0 FF F0 04 C4 FF B0
9038- EA C6 06 A5 06 C9 FF F0
9040- 04 C5 FC B0 D3 60 20 E2
9048- F3 A9 00 8D 52 C0 20 D8
9050- F3 A9 40 85 E6 A9 00 8D
9058- 54 C0 A9 06 85 07 A5 19
9060- 85 FF A9 27 85 FE A9 00
9068- 85 FC A5 ED 85 FD 20 00
9070- 90 A9 00 8D 55 C0 A9 20
9078- 85 E6 A9 07 85 07 20 00
9080- 90 A9 06 85 07 A9 20 85
9088- E6 A9 00 8D 55 C0 20 A0
9090- 90 A9 40 85 E6 A9 00 8D
9098- 54 C0 20 A0 90 4C 85 90
90A0- E6 07 A5 07 C9 02 D0 39
90A8- 20 00 90 A4 09 A6 1F CA
90B0- D0 FD 88 D0 F8 A9 07 85
90B8- 07 A4 EC C6 FE C6 FF 88
90C0- D0 F9 A5 FE C5 EF B0 15
90C8- 18 A5 FE 65 EC 85 1D A5
90D0- FF 65 EC 85 1E A9 27 85
90D8- FE A5 19 85 FF 20 00 90
90E0- 60 A5 07 C9 08 D0 4E A9
90E8- 01 85 07 A4 EC E6 FE E6
90F0- FF 88 D0 F9 A5 FE C5 EB
90F8- D0 0F 38 A5 1D E5 EC 85
9100- 08 A5 1E 85 FF A5 1D 85
9108- FE 20 00 90 A4 09 A6 1F
9110- CA D0 FD 88 D0 F8 A4 EC
9118- C6 FE C6 FF 88 D0 F9 A9
9120- 06 85 07 A5 FE C5 08 D0
9128- 08 A9 27 85 FE A5 19 85
9130- FF 20 00 90 60 20 00 90
9138- A4 09 A6 1F CA D0 FD 88
9140- D0 F8 C6 07 C6 07 20 00
9148- 90 18 AD 00 C0 C9 80 B0
9150- 01 60 68 28 60 00 10 C0
```

電腦時代程式磁碟



第3期程式磁碟

附有程式之文章	在磁碟索引表中之相對檔案名稱
DOS的自動同步追踪器	A DOS SELF-SYNCH DEMO
動量不滅定律模擬法	AIR CAR - MOMENTUM
自製磁碟索引表封套	APPLE DISKETTE ENVELOPE
蘋果編曲器	APPLE MUSIC WRITER
	PROGRAM#1-SOUND & VARIATION
	PROGRAM#2-PADDLE SOUND
	PROGRAM#3.OK
	PROGRAM#4.OK
	PROGRAM#5.OK
幼童認數字初階	BOUNCING NUMBER-MAIN PROGRAM
	MSOUND
	NUMBER TABLE
賽跑遊戲	PATHWAY
步步追踪	CHASE
DOS KILLER	DOS KILLER
圍棋遊戲	DOT
巴哈的古典樂章名作	FUNGUE
旋轉螢幕畫面	LEAVING HOME
	LEAVING HOME DEMO
雜誌目錄分類歸檔系統	MAGAZINE FILE TRACKING SYSTEM
神經病醫生—神經鬆弛體操	RELAX ROUTS
	RELAX TEXT
	RELAXATION-MAIN PROGRAM
	CREATE RELAX FILES

第一期程式磁碟

	附程式之文章	在磁碟索引表中之相對檔案名稱
1	又破傳統的鍵盤	DVORAK KEYBOARD
2	捉球遊戲	BEND OF ROUNDING
3	巧闖電腦室	GAME OF SABOTAGE
4	迷宮教室	GAME OF MAZING
5	將 APPLESOFT 變為 FORTRAN	MAZE SHAPES
		APPLESOFT TO FORTRAN
6	上、下、左、右移動文字畫面	PEELING THE TEXT SCREEN
		PEELING LISTING 5
7	測試工具—BOOLEAN 程式	BOOLEAN TUTORIAL
		BOOLEAN TUTORIAL—MAIN PROGRAM
8	建立多磁碟空間—般的HELLO	GREETING TRANSFER
		GREETING
9	重現刪除之檔案	RETRIEVE THE DELETED PROGRAM
10	電腦時代月刊	APPLE CALENDAR
		APPLE CALENDAR & SUM
		SUM I
		CAL.FILE—1983
		CAL.FILE—1984
		CAL.FILE—1985
		CAL.FILE—1986
11	COMPUTEXT 文字處理	COMPUTEXT
12	時日相距計算程式	DYNAMIC CALENDAR
13	增強英文字彙能力	GAME OF SPELLING TEST
14	幼童數學四則運算練習	GAME OF MATHEMATIC
15	連四棋	GAME OF CONNECT FOUR

第二期程式磁碟

	附有程式之文章	在磁碟索引表中之相對檔案名稱
1	貨櫃處理	CONTAINER HANDLING
2	沙蟹幻方局	POKER
		LOADER
		POKER SHOWHAND
3	巧破密令	GAME OF CRYPTO
4	美術字繪劃器	RETRIEVE FONTASTIC
		SCROLL!!
		SUPERFONTASTIC
		LEAD
		CHAR 1
5	小寫字母處理	LOWER CASE CHIP CONVERTER
6	程式轉載器	SAFELY HANDLING PROGRAM
7	彩色電視／監視器色彩測試	CRT COLOR DEMONSTRATION TEST
8	保留遊戲積分記錄	HIGH SCORE KEEPER
9	數學附件程式	MATHS TOOL KIT 1
		Z
10	磁碟修護工具	BPHTCH
		PATCH2
11	加送電腦時代月刊目錄一覽	LOADSUM 2
		SUM2

歡迎補購 各期電腦時代程式磁碟

請前往COMPUHOUSE補購

地址：深水埗南昌街白田購物中心（近南昌戲院）二樓197室　　電話：3-7776383

第二期「破解密令」有獎遊戲得獎者名單

頭獎：曾永聰（筲箕灣道）獲贈閱「電腦時代」雜誌及程式磁碟十二期。

二獎：十名

(1) 陳君海（石蔭邨）
(2) 陳坤成（大南街）
(3) 潘賓光（美孚新邨）
(4) 羅耀林（崇齡街）
(5) 蘇志權（般含道）
(6) 鍾志強（堅　道）
(7) 劉福來（屯　門）
(8) 梁澤培（灣仔道）
(9) 李漢強（元朗邨）
(10) 李頌文（通洲街）

各獲獎「電腦時代」雜誌十二期。

三獎：三十名

獲贈第五期「電腦時代程式磁碟一隻」。

(1) 朱雪梅（柯士甸道）
(2) 何　勇（山谷道）
(3) 林俊傑（牛頭角）
(4) 梁鈞錫（藍田邨）
(5) 陳華裕（元洲街）
(6) 陳端成（旺角道）
(7) 黃木波（筲箕灣道）
(8) 楊振雄（柴灣道）
(9) 廖國光（橫頭磡）
(10) 趙裕林（福華街）
(11) 劉燦華（石硤尾）
(12) 劉應光（新蒲崗）
(13) 鄭子龍（坪　石）
(14) 鄧顯明（秀茂坪）
(15) 歐陽志威（九龍塘）
(16) 盧錫光（九龍塘）
(17) 劉錦標（天光道）
(18) 黎松安（彌敦道）
(19) 蔡長宗（太子行）
(20) 鄭守成（長　洲）
(21) 鄭啓光（大坑東）
(22) 歐桂泉（金巴倫道）
(23) 蕭秀兒（油塘）
(24) 關冠財（藍田邨）
(25) 譚立民（上海街）
(26) 譚士安（七姊妹道）
(27) 羅健文（偉晴街）
(28) 林錦棠（渡船街）
(29) 胡家祥（元洲街）
(30) 莫永源（荃威花園）

安慰獎：一百名

獲贈閱第五期「電腦時代」一本。

(1) 陳佩蘭（沙田禾輋邨）
(2) 梁耀昌（大同新邨）
(3) 潘國銓（堅道英輝台）
(4) 林嘉明（黃大仙上邨）
(5) 吳嘉駒（樂民新邨）
(6) 鄭華生（牛頭角下邨）
(7) 林漢華（元洲新邨）
(8) 霍德懷（協和街）
(9) 盧明中（石硤尾第九座）
(10) 趙明仁（沙田愉城苑）
(11) 劉振邦（皇后大道西）
(12) 曾錦程（麗閣邨）
(13) 李應虹（美景街）
(14) 劉浩權（藍田第八座）
(15) 鄺燦謙（西環觀龍樓）
(16) 陳國明（北角健威花園）
(17) 王志偉（瀝源邨）
(18) 趙仕培（坪石邨藍石樓）
(19) 江仲權（華富邨華康樓）
(20) 凌偉聰（長洲大新街）
(21) 賴國強（牛頭角上邨）
(22) 鄭鑑釗（亞皆老街）
(23) 陳冠倫（基隆街）
(24) 陳錦活（牛頭角上邨）
(25) 梁錦誠（琼山苑D座）
(26) 程溢民（元朗大馬路）
(27) 梁景豪（長沙灣邨）
(28) 蘇石根（沙田禾輋邨）
(29) 譚志雄（上海街）
(30) 潘克堅（荃灣福來邨）
(31) 茹兆煌（藍田15座）
(32) 陳慧芬（何文田街）
(33) 陳國輝（淺甸山布思道）
(34) 林子文（大南街）
(35) 呂健昌（大南街）
(36) 蕭昇蕃（香港堅道）
(37) 葉鎮隆（樂富邨）
(38) 趙惠慈（筲箕灣太安樓）
(39) 侯肇輝（筲箕灣太安樓）
(40) 湯秀儀（謝非道）
(41) 楊文淵（北角英皇道）
(42) 梁建民（新田圍邨）
(43) 高廣安（大埔宏福苑）
(44) 胡偉強（筲箕灣）
(45) 陳乃寬（土瓜灣）
(46) 黃家輝（梨木樹邨）
(47) 吳偉鴻（華富邨）
(48) 陳松青（葵涌葵盛邨）
(49) 范志偉（坪石邨）
(50) 關家儀（官塘月華街）
(51) 胡志榮（九龍城郵政局）
(52) 歐偉強（九龍白田邨）
(53) 鄭志昌（荃灣石圍角）
(54) 關秉潮（青衣島長青邨）
(55) 鄧國明（太古城）
(56) 鍾國偉（梨木樹）
(57) 劉宗國（官塘月華街）
(58) 趙志輝（沙田禾輋邨）
(59) 文年德（元朗安興路）
(60) 陳廣傑（屯門大興邨）
(61) LEE CHAK MAN
(62) 鄭德昌（九龍白田邨）
(63) 馮詠璇（白田上邨）
(64) 李景棣（黃大仙上邨）
(65) 何嘉仁（香港卑利街）
(66) 歐陽成文（沙田沙角邨）
(67) AU WAI SHANG
(68) CHEONG SHUN FAI
(69) CHAN MING YI
(70) WONG KAI YIN
(71) KO WING HON
(72) KEVIN YU
(73) H.F.LEUNG
(74) MAK MIU SUNG
(75) LI KAM TAI
(76) CHAN NAM KIU
(77) TAM WUN PING
(78) WONG HING SHAN
(79) JENNY TONG
(80) LUM KIN
(81) LEUNG CHI WO
(82) CHENG WAN KWAN
(83) KWOK KA FU
(84) KING-KUEN LAU
(85) AU-YEUNG CHING
(86) WONG SHU CHIU
(87) LAU WAN TO
(88) CHENG YEW KWONG
(89) POON FAI YIU
(90) LEE WAI CHEONG
(91) HO KA SHUN
(92) STANLEY TSUI
(93) TSANG CHUN HUNG
(94) WAN CHI LEUNG
(95) LAU MAN FAI
(96) CHEUNG WAI CHAN
(97) CHAN LOK SHANG
(98) IP CHUN CHIU
(99) YUN CHUN YI
(100) HO WAI MING

應用數學套件程式（二）

計算功能包括：

①分解數值爲質因數
②計算各類三角形邊長及角度
③計算多邊形面積

筆者在電腦時代第二期中編寫了一個數字套件程式給在求學中同學，相信你們會發覺這套件程式相當實用。今期筆者再接再勵，爲讀高小及初中的同學寫出了第二個數學應用套件程式。程式將包括三項運算功能：

（1）將一個輸入數值分解爲質因數。
（2）處理各種三角形邊長及角度問題，包括以 RADIAN 輸出角度的運算結果及以 DEGREE 輸出運算結果。
（3）計算任何多邊形的面積，只要輸入多邊形的角點的座標便可。

執行本程式時，讀者只按入1、2或3隨意選擇計算功能。在第二項處理三角形問題中，若要選擇 RADIAN 輸出答案，可按入"Y"，若是以 DEGREE 做答案，按入任何鍵均可。在計算三角形邊長及角度問題方面，共有五種方式選擇：（A）ASA，（B）SAS，（C）AAS，（D）SSA，（E）SSS 。其中A爲角度（ANGLE），S爲邊長（SIDE）。在第三項計算多邊形面積功能時，必須輸入各角點在X—Y座標中的位置，此可包括負數。■

數學套件應用程式

```
400  INVERSE
500  HOME : VTAB (5): PRINT "MATH
     EMATIC TOOL KIT 2"
501  PRINT
505  NORMAL : HTAB (15): PRINT "O
     PTIONS"
510  HTAB (15): PRINT "1. Prime F
     actor of Number"
511  PRINT
515  HTAB (15): PRINT "2. Triangl
     e Problem"
516  PRINT
520  HTAB (15): PRINT "3. Area of
      Polygon"
550  GET SEL: IF SEL > 3 OR SEL <
     1 THEN  GOTO 505
560  ON SEL GOTO 1000,2000,3000
1000  REM  * PRIME FACTOR *
1005  REM
1010  REM  THIS PROGRAM DETERMINE
     S THE
1015  REM  PRIME FACTORS OF AN IN
     TEGER
1020  REM
1025 Z = 0:S = 0:I = 0
1030  GOTO 1145: REM  MAIN
1035  REM  DISPLAY
1040  IF S = 0 THEN 1060
1045  PRINT I;
1050  PRINT "<";
1055  PRINT S
1060  RETURN
1065  REM  CALCULATE
1070 X = Z
1075  FOR I = 2 TO X
1080 S = 0
1085  IF (Z / I) =  INT (Z / I) THEN
    Z = Z / I:S = S + 1: GOTO 10
    85
1090  GOSUB 1035: REM  DISPLAY
1095  NEXT
1100  RETURN
1105  REM  NUM
1110  PRINT
1115  PRINT "NUMBER: ";
1120  INPUT "";Z
1125  IF Z = 0 THEN 1140
1130  GOSUB 1065: REM  CALC
1135  GOTO 1110
1140  RETURN
1145  REM  MAIN
1150  HOME
1155  PRINT "PRIME FACTOR ROUTINE
     "
1160  PRINT "ENTER A ZERO TO END"

1165  GOSUB 1105: REM  NUM
1170  PRINT
1175  PRINT
1180  PRINT "GOODBYE"
1185  END
2000  HOME : PRINT  TAB( 11)"PART
     S OF A TRIANGLE"
2003  PRINT : PRINT  TAB( 11)"ANG
     LES IN RADIANS (Y/N) ?";: GET
     ANS$: IF ANS$ <  > "Y" THEN
     2350
2005  PRINT
2010  DIM A(3),S(3)
2015  REM  SET VALUE OF PI
2020 P = 3.14159265
2025  REM  SET CONVERSION FACTOR
     FOR COMVERTING DEGREES TO RA
     DIANS
2030  REM  OF THE TRIANGLE WHERE
     A=AGLE, S=LENGTH OF SIDE
2035  PRINT "PROBLEM TYPES: 1=ASA
     , 2=SAS, 3=AAS "
2036  PRINT : PRINT "
       4=SSA, 5=SSS, 6=END"
2040  PRINT : INPUT "PROBLEM TYPE
     : ";X
2045  REM  DIRECT PROGRAM TO PROP
     ER CALCULATIONS
2050  ON X GOTO 2070,2100,2135,21
```

```
    55,2200,2300
2055  REM  INVALID ENTRY
2060  PRINT  CHR$ (7);" INVALID T
     YPE": PRINT : GOTO 2035
2065  REM  TYPE 1
2070  INPUT "ENTER ANGLE,SIDE,ANG
     LE,SIDE,ANGLE: ";A(1),S(3),A
     (2)
2075 A(3) = P - A(1) - A(2)
2080 S(1) = S(3) *  SIN (A(1)) /
      SIN (A(3))
2085 S(2) = S(3) *  SIN (A(2)) /
      SIN (A(3))
2090  GOTO 2220
2095  REM  TYPE 2
2100  INPUT "SIDE,ANGLE,SIDE: ";S
     (3),A(1),S(2)
2105 S(1) =  SQR (S(3) ^ 2 + S(2)
      ^ 2 - 2 * S(3) * S(2) *  COS
     (A(1)))
2110 A(2) =  SIN (A(1)) / S(1) *
     S(2)
2115 A(2) =  ATN (A(2) /  SQR ( -
     A(2) * A(2) + 1))
2120 A(3) = P - A(1) - A(2)
2125  GOTO 2220
2130  REM   TYPE 3
2135  INPUT "ENTER ANGLE,ANGLE,SI
     DE: ";A(3),A(2),S(3)
2140 A(1) = P - A(2) - A(3)
2145  GOTO 2080
2150  REM   TYPE 4
2155  INPUT "SIDE,SIDE,ANGLE: ";S
     (1),S(2),A(1)
2160 T = S(2) *  SIN (A(1))
2165  IF S(1) < T THEN 2285
2170 S(3) =  SQR (S(2) ^ 2 - T ^
     2)
2175  IF S(1) <  = T THEN 2190
2180 Y =  SQR (S(1) ^ 2 - T ^ 2)
2185 S(3) = S(3) + Y
2190  GOTO 2110
2195  REM  TYPE 5
2200  INPUT "ENTER SIDE,SIDE,SIDE
     : ";S(1),S(2),S(3)
2205 A(1) = (S(2) ^ 2 + S(3) ^ 2 -
     S(1) ^ 2) / 2 / S(2) / S(3)
2210 A(1) =  -  ATN (A(1) /  SQR
     ( - A(1) * A(1) + 1)) + 1.57
     08
2215  GOTO 2110
2220  PRINT
2225  REM  PRINT RESULTS
2230  PRINT : FOR I = 1 TO 3
2235  REM  THE ANGLE OF A TRIANGL
     E CONNOT BE LESS THAN 0
2240  IF A(I) < 0 THEN 2285
2245  PRINT "SIDE ";I;" = "; INT
     (S(I) * 1000 + .5) / 1000
2250  PRINT "OPPOSITE ANGLE = "; INT
     (A(I) * 1000 + .5) / 1000;"
     Radians"
2255  NEXT I
2260  PRINT
2265  REM  RESTART PROGRAM
2270  INPUT "PRESS<RETURN> FOR NE
     XT PROBLEM ";Z$
2275  REM  CLEAR SCREEN FROM LINE
      5 DOWN
2280  VTAB 8: CALL  - 958: GOTO 2
     040
2285  PRINT "NO SOLUTION"
2290  REM  COMPLETE LOOP
2295 I = 3: GOTO 2255
2300  END
2350  HOME : PRINT  TAB( 11)"PART
     S OF A TRIANGLE"
2355  PRINT
2360  DIM A(3),S(3)
2365  REM  SET VALUE OF PI
2370 P = 3.145927
2375  REM  SET CONVERSION FACTOR
     FOR COMVERTING DEGREES TO RA
     DIANS
2380 C = .0174532927
2385  REM  OF THE TRIANGLE WHERE
     A=AGLE, S=LENGTH OF SIDE
2390  PRINT "PROBLEM TYPES: 1=ASA
     , 2=SAS, 3=AAS"
2391  PRINT : PRINT "
       4=SSA, 5=SSS, 6=END"
2395  PRINT : INPUT "PROBLEM TYPE
     : ";X
2400  REM  DIRECT PROGRAM TO PROP
     ER CALCULATIONS
2405  ON X GOTO 2425,2465,2505,25
     35,2585,2685
2410  REM  INVALID ENTRY
2415  PRINT  CHR$ (7);" INVALID T
     YPE": PRINT : GOTO 2390
2420  REM  TYPE 1
2425  INPUT "ENTER ANGLE,SIDE,ANG
     LE,SIDE,ANGLE: ";A(1),S(3),A
     (2)
```

```
2430 A(1) = A(1) * C
2435 A(2) = A(2) * C
2440 A(3) = P - A(1) - A(2)
2445 S(1) = S(3) *  SIN (A(1)) /
     SIN (A(3))
2450 S(2) = S(3) *  SIN (A(2)) /
     SIN (A(3))
2455  GOTO 2605
2460  REM  TYPE 2
2465  INPUT "SIDE,ANGLE,SIDE: ";S
    (3),A(1),S(2)
2470 A(1) = A(1) * C
2475 S(1) =  SQR (S(3) ^ 2 + S(2)
     ^ 2 - 2 * S(3) * S(2) *  COS
    (A(1)))
2480 A(2) =  SIN (A(1)) / S(1) *
    S(2)
2485 A(2) =  ATN (A(2) /  SQR ( -
    A(2) * A(2) + 1))
2490 A(3) = P - A(1) - A(2)
2495  GOTO 2605
2500  REM    TYPE 3
2505  INPUT "ENTER ANGLE,ANGLE,SI
    DE: ";A(3),A(2),S(3)
2510 A(3) = A(3) * C
2515 A(2) = A(2) * C
2520 A(1) = P - A(2) - A(3)
2525  GOTO 2445
2530  REM   TYPE 4
2535  INPUT "SIDE,SIDE,ANGLE: ";S
    (1),S(2),A(1)
2540 A(1) = A(1) * C
2545 T = S(2) *  SIN (A(1))
2550  IF S(1) < T THEN 2670
2555 S(3) =  SQR (S(2) ^ 2 - T ^
    2)
2560  IF S(1) <  = T THEN 2575
2565 Y =  SQR (S(1) ^ 2 - T ^ 2)
2570 S(3) = S(3) + Y
2575  GOTO 2480
2580  REM  TYPE 5
2585  INPUT "ENTER SIDE,SIDE,SIDE
    : ";S(1),S(2),S(3)
2590 A(1) = (S(2) ^ 2 + S(3) ^ 2 -
    S(1) ^ 2) / 2 / S(2) / S(3)
2595 A(1) =  -  ATN (A(1) /  SQR
    ( - A(1) * A(1) + 1)) + 1.57
    08
2600  GOTO 2480
2605  PRINT
2610  REM  PRINT RESULTS
2615  PRINT : FOR I = 1 TO 3
2620  REM  THE ANGLE OF A TRIANGL
    E CONNOT BE LESS THAN 0
2625  IF A(I) < 0 THEN 2670
2630  PRINT "SIDE ";I;" = "; INT
    (S(I) * 1000 + .5) / 1000
2635  PRINT "OPPOSITE ANGLE = "; INT
    (A(I) / C * 1000 + .5) / 100
    0;" Degrees"
2640  NEXT I
2645  PRINT
2650  REM  RESTART PROGRAM
2655  INPUT "PRESS<RETURN> FOR NE
    XT PROBLEM ";Z$
2660  REM  CLEAR SCREEN FROM LINE
     5 DOWN
2665  VTAB 6: CALL  - 958: GOTO 2
    395
2670  PRINT "NO SOLUTION"
2675  REM  COMPLETE LOOP
2680 I = 3: GOTO 2640
2685  END
3000  HOME : PRINT  TAB( 12)"AREA
     OF A POLYGON"
3005  PRINT
3010  INPUT "NUMBER OF VERTICES :
     ";N
3015  PRINT : IF N < 1 THEN 3105
3020  DIM X(N + 1),Y(N + 1)
3025  FOR I = 1 TO N
3030  IF I > 1 THEN 3045
3035  PRINT "COORDINATES OF VERTE
    X ";I;
3040  GOTO 3050
3045  PRINT "                 VERTE
    X ";I;
3050  HTAB 25: INPUT " (A,B)  :";
    X(I),Y(I)
3055  NEXT I
3060 X(N + 1) = X(1)
3065 Y(N + 1) = Y(1)
3070 A = 0
3075  FOR I = 1 TO N
3080 A = A + (X(I) + X(I + 1)) *
    (Y(I) - Y(I + 1))
3085  NEXT I
3090  PRINT : PRINT "AREA= "; ABS
    (A) / 2
3095  PRINT
3100  INPUT "PRESS 'RETURN' TO RE
    START PROGRAM ";Z$: CLEAR : GOTO
    3000
3105  END
```

英文串字練習

加深兒童認字能力

一向以來，設計給學前兒童學習英文、英文串字的遊戲並不多，很容易玩膩，因此我們不斷推陳出新，務求達到有足夠的程式來滿足他們。

本程式是顯示一些三個字母組成的英文生字在螢幕上，要求兒童照樣按入，當按對了便會在揚聲器上發出三個音符以示鼓勵。按對次數越多，獎勵的音樂維持越長。

所有的英文生字是存放在行號 1080 的 DATA 句中，各位家長可視乎兒童的程度來作修改，增加生字的難度。最後離開本程式時，只須按下 ESC 鍵便可以了。

編者註：由 COMPUHOUSE 製作的本期程式磁碟中，「英文串字練習」程式中加插了兩幅高解像圖形——唐老鴨及小肥豬。若小孩子按對了生字，電腦會閃現出這兩幅圖像，以示獎勵。在執行主程式時，這兩個圖像會自動載入。□

英文串字練習程式

```
1000  REM  ********************
1010  REM  *                  *
1020  REM  *   LOVING BEAR    *
1030  REM  *                  *
1035  REM  *       BY         *
1037  REM  *   COMPUHOUSE     *
1038  REM  *                  *
1040  REM  ********************
1045  REM
1050  HOME : HGR : VTAB 10: HTAB
     10: PRINT "HI KATIE": TEXT
1053  PRINT  CHR$ (4);"BLOAD BEAR
     1": GOSUB 6000: VTAB 24: PRINT
     "HOW ARE YOU?"
1054  PRINT  CHR$ (4);"BLOAD BEAR
     2,A$4000"
1055  DIM A$(100)
1060  HOME : VTAB 10: HTAB 13: PRINT
     "HERE WE GO": TEXT
1065  PRINT  CHR$ (4);"BLOAD TOOT
     LE"
1070  READ JJ: FOR J = 1 TO JJ: READ
     A$(J): NEXT J
1075  POKE  - 16302,0
1080  DATA  28
1090  DATA   "CAT","DOG","BIG","E
     AT","RUN","SHE","SUN","THE",
     "AND","GO","OFF","HIS","HER"
     ,"YOU","FOR","WILL","NOT","W
     AS","RED","HAD","RUN","SEE"
1100  DATA  "SIT","SAW","WHY","WE
     T","YES","COW"
1200  POKE 7,3:K = 0
1210  TEXT : HOME :K = K + 1
1220  VTAB 10: HTAB 18
1230 J =  INT (JJ *  RND (1)) + 1

1240  PRINT A$(J)
1250 I = 1:II =  LEN (A$(J))
1260  POKE 6,75: CALL 806: REM  T
     WEET
1265  GOSUB 5000: REM  DELAY
1270  VTAB 12: HTAB 17 + I: PRINT
     "?": GET B$
1280  IF B$ <  >  MID$ (A$(J),I,1
     ) THEN 1330
1290  CALL 806: REM  TWEET
1300  VTAB 12: HTAB 17 + I: PRINT
     B$
1310 I = I + 1: IF I > II THEN 13
     60
1320  GOTO 1270: REM  NEXT WORD
1330  POKE 6,64: CALL 806: REM  W
     RONG!
1340  POKE 6,75
1350  GOTO 1270: REM  GET ANOTHER
     LETTER
1360  GOSUB 5000
1362  IF K > 10 THEN 1365
1364  IF 10 *  RND (1) + 1 > 6 THEN
     GOSUB 6000: GOSUB 5000: TEXT

1365  ON K / 5 + 1 GOSUB 2000,300
     0,4000
1367  IF K >  = 15 THEN  GOSUB 40
     00
1370  GOTO 1210: REM  NEXT WORD
1997  REM
1998  REM
1999  REM  REWARD 1
2000  POKE 6,84: CALL 806: GOSUB
     5000: RETURN
2997  REM
2998  REM
2999  REM  REWARD 2
3000  POKE 6,103: CALL 806: GOSUB
     5000: RETURN
3997  REM
3998  REM
3999  REM  REWARD 3
4000  IF 10 *  RND (1) + 1 > 6 THEN
     GOSUB 7000
4020  POKE 6,132: CALL 806: GOSUB
     5000: RETURN
4997  REM
4998  REM
4999  REM  DELAY
5000  FOR L = 1 TO 300: NEXT L: RETURN

5997  REM
5998  REM
5999  REM  DISPLAY SCREEN 1
6000  POKE  - 16300,0: POKE  - 16
     297,0: POKE  - 16304,0: RETURN

6997  REM
6998  REM
6999  REM  WINK
7000  GOSUB 6000: GOSUB 5000: POKE
     49237,1: GOSUB 5000: POKE 49
     236,1: RETURN
```

■

豐富設計的學前兒童教育程式

由A至Z、到0至9；加、減、大、小概念盡在其中

這是一個有豐富設計的程式，對於學前孩童必定非常吸引。程式會以許多生動有趣圖案，來描繪A至Z及0至9英文字母數字以加強兒童對他們的印象，此外尚有一些基本的概念，例如大、小、加、減等。

如果程式是在彩色電視下進行，則更爲吸引。

執行程式後，只要按入鍵盤上任何鍵子，都會有相對應的生動圖形及動聽的音樂出現。

各位家長，有空時不妨坐下來陪陪你的孩子一同玩此遊戲。

以下是各個按鍵會出現的畫面圖案：

A——蘋果（APPLE）
B——熊人（BEAR）
C——貓（CAT）
D——狗（DOG）
E——空的概念（EMPTY）
F——農莊（FARM）
　　當連續按第二次鍵時，會隨機出現各種的動物。
H——馬（HORSE）
I——雪糕筒（ICE CREAM）
　　包括有：雲呢嗱、車厘子、朱古力、士多啤梨及朱古力批。
J——跳躍（JUMP）
K——澳洲浣熊（KOALA WALLY）
L——羊（LAMB）
M——磁石（MAGNET）
N——數目（NUMBER）
　　重覆按N鍵會不斷將數字加添上。
O——會出現一首耳熟能詳的調子。
P——金字塔（PRISM）
Q——快快行及慢步的毛蟲。
R——白兎（RABBIT）
S——星星（STAR）配托有音樂背景
U——上（UP）及下（DOWN）
V——火山（VOLCANO）
W——毛蟲（WORM）
X——大X
Z——迂迴曲折（ZIG—ZAG）
←——由右至左的概念。
→——由左至右的概念。
?——由上至下的概念。
*——由下至上的概念。
0至9——音符DO、RE、ME
+——相加的概念。
=——相等的概念。
-——不相等的概念。
>——顯示大於的概念。
<——顯示細於的概念。
ESC——離開本程式。
RETURN——清抹畫面。

程式藏有許多不同的變化，千萬不要看輕它。最重要是多次重覆按下鍵掣，便會有意想不到的變化出現。■

```
]LIST

8 TW = 770:WN = 769:ZR = 768
9  DEF  FN RN(RV) =  INT ( RND (1) * R
   V)
10  DIM CS(10),TN(30),DURV(30)
11  DIM WR(20),USED(20)
12  GOSUB 50:FLAG = 0:CUT = 0
16 BPCH = 190:BDUR = 100:DRINC = 50
18 BSTP = 10:B2 = 15:P2 = BPCH - 5 * B
   STP
19  GOSUB 4350
20  GOTO 6600
21  PLOT 5,BOT + 2: POKE ZR,BP: POKE W
   N,BD: CALL TW: COLOR= 0: PLOT 5,BO
   T + 2: RETURN
22  PLOT 5 + NT,BOT + 2: POKE ZR,BP -
   2 * BSTP: POKE WN,BDUR: CALL TW: COLOR=
   0: PLOT 5 + NT,BOT + 2: RETURN
23  PLOT 5 + 2 * NT,BOT + 2: POKE ZR,B
   P - 4 * BS: POKE WN,BDUR: CALL TW:
    COLOR= 0: PLOT 5 + 2 * NT.BOT + 2
   : RETURN
24  PLOT 5 + 3 * NT,BOT + 2: POKE ZR,B
   P - 5 * BS: POKE WN,BDUR: CALL TW:
    COLOR= 0: PLOT 5 + 3 * NT,BOT + 2
   : RETURN
25  PLOT 5 + 4 * NT,BOT + 2: POKE ZR,P
   2 - B2: POKE WN,BDUR: CALL TW: COLOR=
   0: PLOT 5 + 4 * NT,BOT + 2: RETURN

26  PLOT 5 + 5 * NT,BOT + 2: POKE ZR,P
   2 - 2 * B2: POKE WN,BDUR: CALL TW:
    COLOR= 0: PLOT 5 + 5 * NT,BOT + 2
   : RETURN
27  PLOT 5 + 6 * NT,BOT + 2: POKE ZR,P
   2 - 3 * B2 + B2 / 3: POKE WN,BDUR:
    CALL TW: COLOR= 0: PLOT 5 + 6 * N
   T,BOT + 2: RETURN
28  PLOT 5 + 7 * NT,BOT + 2: POKE ZR,P
   2 - 3 * B2: POKE WN,BDUR: CALL TW:
    COLOR= 0: PLOT 5 + 7 * NT,BOT + 2
   : RETURN
29  GOTO 2345
30 OO = 1
31 R = 10
32  FOR Z = X TO X + L STEP SP
33  PLOT Z,Y
34 P = 30 + Z: POKE ZR,P: POKE WN,R
35  CALL TW
36  FOR J = 1 TO D: NEXT J
37  NEXT Z
39  RETURN
40  GOSUB 8100
41 R = 10
42  FOR Z = Y TO Y + L STEP SP
43  PLOT X,Z
44 P = 30 + Z: POKE ZR,P: POKE WN,R
45  CALL TW
46  FOR J = 1 TO D: NEXT J
47  NEXT Z
49  RETURN
50  POKE TW,173: POKE TW + 1,48: POKE
   TW + 2,192: POKE TW + 3,136: POKE
   TW + 4,208
51  POKE TW + 5,5: POKE TW + 6,206: POKE
```

```
   TW + 7,1: POKE TW + 8,3
52  POKE TW + 9,240: POKE TW + 10,9: POKE
   TW + 11,202: POKE TW + 12,208
53  POKE TW + 13,245: POKE TW + 14,175
   : POKE TW + 15,0: POKE TW + 16,3
54  POKE TW + 17,76: POKE TW + 18,2: POKE
   TW + 19,3: POKE TW + 20,96
55  RETURN
60  WAIT 49152,128,1:P =  PEEK ( - 163
   84): POKE  - 16368,0
61  IF P = 141 OR P = 155 THEN 69
62  IF P ) 184 OR P ( 177 THEN 60
63  COLOR= 1:P = P - 176
64  ON P GOSUB 21,22,23,24,25,26,27,28

65  GOTO 60
69 FLAG = 0: GOTO 100
70  GR
71  COLOR= 15:TP = 12:BOT = 32:NT = 4
72  FOR I = TP TO BOT - NT STEP NT: HLIN
   0,39 AT I: NEXT I
73  VLIN TP,BOT - NT AT 0
74  RETURN
75 OO = 1
77  FOR I = LOW - WID TO LOW
78  HLIN LEF,LEF + WID - 1 AT I
79  NEXT I
80  VLIN LOW - 3 * WID,LOW - WID AT LE
   F + WID - 1
81  RETURN
82 SV = BDUR:LEF = 0
83  FOR IX = 1 TO XLN
84 BDUR = DURV(IX)
85  ON TN(IX) GOSUB 1075,1125,1175,122
   5,1275,1325,1375,1425
86  NEXT IX
87 BDUR = SV
88  RETURN
90  CALL  - 936: GR
95 FGJ = 0
100 CUT = 0: WAIT 49152,128,1:P =  PEEK
   ( - 16384)
110  POKE  - 16368,0
120  IF P (  = 127 THEN 100
121  IF P = 206 AND FGJ = 1 THEN 2312:
   FGJ = 0
122  CALL  - 936: COLOR=  FN RN(15) +
   1:FGJ = 0
123  IF P = 160 THEN 135
124  IF P = 141 THEN 90
125  IF P = 136 THEN 170
130  IF P = 155 THEN 3200
131  IF P = 149 THEN 140
132  IF P ( 160 THEN 2500
133  IF P ) 218 THEN 200
134  GR
135  GOTO 8000
140 OO = 1
145 Y = 8 +  FN RN(20)
150 X = 3:L = 27:D = 20:SP = 3
155  GOSUB 30: GOTO 100
170 OO = 1
175 Y = 5 +  FN RN(20)
180 X = 30:L =  - 27:SP =  - 3:D = 20
185  GOSUB 30: GOTO 100
200 X =  FN RN(30):Y =  FN RN(30):L =
   10
205  HLIN X,X + L AT Y
210  GOSUB 8100
215  VLIN Y,Y + L AT X
220  GOSUB 8100
225  HLIN X,X + L AT Y + L
230  GOSUB 8100
235  VLIN Y,Y + L AT X + L
240  GOTO 100
250 OO = 1
254  GOSUB 8100
255 X = 10:Y = 15: GOSUB 6300
256 X = 16: GOSUB 6290
257 X = 23: GOSUB 6200
258  GOTO 100
260  FOR J = 1 TO 10:DURV(J) = BDUR: NEXT
   J
262 DURV(1) = BDUR + 2 * DRINC:DURV(2)
    = BDUR + 2 * DRINC:DURV(4) = BDUR
    + 2 * DRINC
263 DURV(10) = BDUR + 2 * DRINC
264 TN(1) = 4:TN(2) = 4:TN(3) = 6:TN(4
   ) = 6
265 TN(5) = 6:TN(6) = 5:TN(7) = 5:TN(8
   ) = 3
266 TN(9) = 1:TN(10) = 2
268 XLN = 10: GOSUB 82
270  RETURN
272 TN(1) = 1:TN(2) = 8:TN(3) = 6:TN(4
   ) = 5:TN(5) = 3:TN(6) = 5:TN(7) =
   2
274  FOR I = 8 TO 13:TN(I) = TN(I - 7)
   : NEXT I
276  FOR J = 2 TO 12:DURV(J) = BDUR: NEXT
   J
277 DURV(1) = BDUR + 2 * DRINC:DURV(13
   ) = BDUR + 3 * DRINC
278 DURV(6) = BDUR + 3 * DRINC:DURV(7)
    = BDUR + 3 * DRINC:DURV(8) = BDUR
    + 2 * DRINC
280 XLN = 13: GOSUB 82
282  RETURN
284 XLN = 18
300 OO = 1
301  GR
302  GOSUB 8100
303 X = 10:Y = 15: GOSUB 6350
305 X = 16: GOSUB 6380
307 X = X + 6: GOSUB 6300
308  GOTO 100
310 TN(1) = 1:TN(2) = 1:TN(3) = 5:TN(4
   ) = 5:TN(5) = 6:TN(6) = 6:TN(7) =
   5
312 TN(8) = 4:TN(9) = 4:TN(10) = 3:TN(
   11) = 3:TN(12) = 2:TN(13) = 2:TN(1
   4) = 1
314 XLN = 14
316  FOR I = 1 TO 13:DURV(I) = BDUR: NEXT
   I
318 DURV(7) = BDUR + 2 * DRINC:DURV(14
   ) = BDUR + 2 * DRINC
320  GOSUB 82
322  RETURN
324 XLN = 10
326  FOR I = 1 TO 3:T326 FOR I = 1 TO
   3:TN(I) = 1: NEXT I
328 TN(4) = 2:TN(5) = 3:TN(6) = 3:TN(7
   ) = 2:TN(8) = 3:TN(9) = 4:TN(10) =
   5
330  FOR I = 3 TO 9:DURV(I) = BDUR: NEXT
   I
331 DURV(4) = BDUR - DRINC:DURV(7) = B
   DUR - DRINC
332 B = BDUR + 2 * DRINC:DURV(1) = B:D
   URV(2) = B:DURV(5) = B:DURV(10) =
   B
333 DURV(9) = BDUR - DRINC
334  GOSUB 82
336  RETURN
338 XLN = 10
340 TN(1) = 6:TN(2) = 8:TN(3) = 6:TN(4
   ) = 4:TN(5) = 3:TN(6) = 4
341 TN(7) = 5:TN(8) = 5:TN(9) = 3:TN(1
   0) = 1
343  FOR IP = 1 TO 10:DURV(IP) = BDUR:
    NEXT IP
345 DURV(2) = BDUR + 2 * DRINC:DURV(10
   ) = BDUR + DRINC
347  GOSUB 82
349  RETURN
350  GOTO 5400
352 XLN = 14
354 TN(1) = 4:TN(2) = 4:TN(3) = 5:TN(4
   ) = 6:TN(5) = 4:TN(6) = 6:TN(7) =
   5:TN(8) = 1
356 TN(9) = 4:TN(10) = 4:TN(11) = 5:TN
   (12) = 6:TN(13) = 4:TN(14) = 3
358  FOR Y = 1 TO 12:DURV(Y) = BDUR: NEXT
   Y
360 DURV(13) = BDUR + 2 * DRINC:DURV(1
   4) = BDUR + 2 * DRINC
362  GOSUB 82
364  RETURN
400  GR
402  GR
404  GOSUB 8100
406 X = 10:Y = 15: GOSUB 6210
408 X = X + 5: GOSUB 6300
410 X = X + 5: GOSUB 6360
412 X = X + 5: GOSUB 6330
414  GOTO 100
450  GR : GOSUB 8100
454 X = 10:Y = 15: GOSUB 6210
456 X = X + 5: GOSUB 6240
458 X = X + 6: GOSUB 6370
460 X = X + 6: GOSUB 6200
462  GOTO 100
500  GOSUB 8100
506 X = 12:Y = 15: GOSUB 6340
508 X = X + 5: GOSUB 6240
510 X = X + 6: GOSUB 6390
512  GOTO 100
550  GOSUB 8100
556 X = 7:Y = 15: GOSUB 6340
558 X = X + 5: GOSUB 6200
560 X = X + 5: GOSUB 6370
562 X = X + 6: GOSUB 6200
564 X = X + 5: GOSUB 6290
566  GOTO 100
600  GOSUB 8100
606 X = 7:Y = 15: GOSUB 6200
608 X = X + 5: GOSUB 6240
610 X = X + 6: GOSUB 6220
612 X = X + 5: GOSUB 6230
614 X = X + 5: GOSUB 6350
616  GOTO 100
650  GOSUB 8100
656 X = 9:Y = 15: GOSUB 6290
658 X = X + 6: GOSUB 6240
660 X = X + 6: GOSUB 6290
662 X = X + 6: GOSUB 6200
664  GOTO 100
700  GOTO 1901
750  GOTO 1550
800  GOTO 1600
850  GR
852  GOSUB 8100
```

```
854 NN = 2 * (1 +  FN RN(4))
856 N1 =  FN RN(NN - 1) + 1
858 N2 = NN - N1
859  GOSUB 860: GOTO 870
860  FOR I = 1 TO NN STEP 2
862  VLIN 20,30 AT 20 - I
864  VLIN 20,30 AT 20 + I
866  NEXT I
868  RETURN
870  FOR J = 1 TO 300: NEXT J
872 LT = 20 - NN + 1:RT = 20 + NN - 1
873 LS = 0:RS = 39
874  FOR I = 1 TO N1
876  COLOR= 0: VLIN 20,30 AT LT
878 LT = LT + 2: COLOR= 8
879  VLIN 20,30 AT LS:LS = LS + 2: POKE
    ZR,200 - LS: POKE WN,50: CALL TW
880  FOR J = 1 TO 200: NEXT J
881  NEXT I
882  FOR I = 1 TO N2
884  COLOR= 0: VLIN 20,30 AT RT
886 RT = RT - 2: COLOR= 7
887  VLIN 20,30 AT RS:RS = RS - 2: POKE
    ZR,200 - RS: POKE WN,50: CALL TW
888  FOR J = 1 TO 200: NEXT J
889  NEXT I
891  FOR J = 1 TO 200: NEXT J
892  COLOR= 12: HLIN 18,22 AT 24
893  HLIN 18,22 AT 26:TN(1) = 4:TN(2) =
    6:DURV(1) = 50:DURV(2) = 150:XLN =
    2: GOSUB 82
894  IF N1 = N2 THEN 100
895  COLOR= 1: FOR J = 1 TO 50: NEXT J

896  FOR I = 17 TO 23
897  PLOT I,I + 5
898  NEXT I
899 TN(1) = 1:DURV(1) = 150:XLN = 1: GOSUB
    82: GOTO 100
900  GOTO 1700
921  GR : COLOR= 8
950  GOTO 2502
951  COLOR= 1
955 X =  FN RN(25) + 5:Y =  FN RN(25) +
    7
957  HLIN X + 2,X + 10 - 2 AT Y
958  HLIN X + 2,X + 10 - 2 AT Y + 1
959  HLIN X + 2,X + 10 - 2 AT Y - 1
960  FOR I = 2 TO 4
965  HLIN X + I,X + 10 - I AT Y + I
967  HLIN X + I,X + 10 - I AT Y - I
970  NEXT I
975  COLOR= 8
980  VLIN Y - 7,Y - 5 AT X + 5
985  COLOR= 4: PLOT X + 4,Y - 7: PLOT
    X + 6,Y - 6
990  GOTO 100
1000 OO = 1
1002  GOSUB 70
1004  CALL  - 936
1005 FLAG = 1
1007  COLOR= 6
1009  GOSUB 1060: GOSUB 1110: GOSUB 11
    60
1011  GOSUB 1210: GOSUB 1260: GOSUB 13
    10
1013  GOSUB 1360: GOSUB 1410
1015  GOTO 60
1024 FLAG = 0: GOTO 100
1030  REM   - FIX STAFF
1032  COLOR= 15: FOR I = TP TO BOT - N
    T STEP NT
1033  HLIN LEF,LEF + WID AT I
1034  NEXT I
1035  RETURN
1050 OO = 1
1051  GOSUB 70
1052 C = 1: COLOR= C: GOSUB 1060
1053  GOSUB 1030
1054 C = 6: COLOR= C: GOSUB 1060
1056  IF FLAG = 1 THEN 1015
1058  GOTO 100
1060 OO = 1
1061 LOW = BOT + 1:LEF = 5:WID = 2: GOSUB
    75
1062  COLOR= 15: PLOT 4,BOT: PLOT 5 +
    WID,BOT: COLOR= 6
1063  IF FLAG = 1 OR C = 1 THEN  GOSUB
    1075
1065  RETURN
1075 OO = 1
1076  HTAB LEF + 1: PRINT "DO";
1077  POKE ZR,BP: POKE WN,BD: CALL TW
1079  RETURN
1100 OO = 1
1101  GOSUB 70
1102 C = 1: COLOR= C: GOSUB 1110
1103  GOSUB 1030
1104 C = 6: COLOR= C: GOSUB 1110
1106  IF FLAG = 0 THEN 100
1108  GOTO 1015
1110  REM   - PARAMS
1111 LOW = BOT - 1:WID = NT / 2:LEF =
    5 + 2 * WID: GOSUB 75
1113  IF FLAG = 1 OR C = 1 THEN  GOSUB
    1125
1115  RETURN
1125 OO = 1
1126  HTAB LEF + 1: PRINT "RE";
1127  POKE ZR,BP - 2 * BSTP: POKE WN,B
    DUR: CALL TW
1128  RETURN
1150 OO = 1
1151  GOSUB 70
1152 C = 1: COLOR= C: GOSUB 1160
1154 C = 6: COLOR= C: GOSUB 1160
1156  IF FLAG = 1 THEN 1015
1158  GOTO 100
1160 OO = 1
1161 LOW = BOT - NT + 1:WID = NT / 2:L
    EF = 5 + 4 * WID: GOSUB 75
1163  IF FLAG = 1 OR C = 1 THEN  GOSUB
    1175
1165  RETURN
1175 OO = 1
1176  HTAB LEF + 1: PRINT "MI";
1177  POKE ZR,BPCH - 4 * BSTP: POKE WN
    ,BDUR: CALL TW
1179  RETURN
1200 OO = 1
1201  GOSUB 70
1202 C = 1: COLOR= C: GOSUB 1210
1203  GOSUB 1030
1204 C = 6: COLOR= C: GOSUB 1210
1206  IF FLAG = 0 THEN 100
1208  GOTO 1015
1210 OO = 1
1211 LOW = BOT - NT - 1:WID = NT / 2:L
    EF = 5 + 6 * WID: GOSUB 75
1213  IF FLAG = 1 OR C = 1 THEN  GOSUB
    1225
1215  RETURN
1225 OO = 1
1226  HTAB LEF + 1: PRINT "FA";
1227  POKE ZR,BP - 5 * BSTP: POKE WN,B
    DUR: CALL TW
1228  RETURN
1250 OO = 1
1251  GOSUB 70
1252 C = 1: COLOR= C: GOSUB 1260
1254 C = 6: COLOR= C: GOSUB 1260
1256  IF FLAG = 1 THEN 1015
1258  GOTO 100
1260 OO = 1
1261 LOW = BOT - 2 * NT + 1:WID = NT /
    2:LEF = 5 + 8 * WID: GOSUB 75
1263  IF FLAG = 1 OR C = 1 THEN  GOSUB
    1275
1265  RETURN
1275 OO = 1
1276  HTAB LEF + 1: PRINT "SO";
1277  POKE ZR,P2 - B2: POKE WN,BDUR: CALL
    TW
1279  RETURN
1300 OO = 1
1301  GOSUB 70
1302 C = 1: COLOR= C: GOSUB 1310
1303  GOSUB 1030
1304 C = 6: COLOR= C: GOSUB 1310
1306  IF FLAG = 0 THEN 100
1308  GOTO 1015
1310  REM    - PARAMS
1311 LOW = BOT - 2 * NT - 1:WID = NT /
    2:LEF = 5 + 10 * WID: GOSUB 75
1313  IF FLAG = 1 OR C = 1 THEN  GOSUB
    1325
1315  RETURN
1325  REM    -LABEL
1326  HTAB LEF + 1: PRINT "LA";
1327  POKE ZR,P2 - 2 * B2: POKE WN,BDU
    R: CALL TW
1329  RETURN
1350 OO = 1
1351  GOSUB 70
1352 C = 1: COLOR= C: GOSUB 1360
1354 C = 6: COLOR= C: GOSUB 1360
1356  IF FLAG = 1 THEN 1015
1358  GOTO 100
1360 OO = 1
1361 LOW = BOT - 3 * NT + 1:WID = NT /
    2:LEF = 5 + 12 * WID: GOSUB 75
1363  IF FLAG = 1 OR C = 1 THEN  GOSUB
    1375
1365  RETURN
1375 OO = 1
1376  HTAB LEF + 1: PRINT "TI";
1377  POKE ZR,P2 - 3 * B2T2 + B2 / 3: POKE
    WN,BD: CALL TW
1379  RETURN
1400 OO = 1
1401  GOSUB 70
1402 C = 1: COLOR= C: GOSUB 1410
1403  GOSUB 1030
1404 C = 6: COLOR= C: GOSUB 1410
1406  IF FLAG = 0 THEN 100
1408  GOTO 1015
1410  REM     - PARAMS
1411 LOW = BOT - 3 * NT - 1:WID = NT /
    2:LEF = 5 + 14 * WID: GOSUB 75
1413  IF FLAG = 1 OR C = 1 THEN  GOSUB
    1425
1415  RETURN
1424  GOTO 100
```

豐富設計的學前兒童教育程式

```
1425 OO = 1
1426  HTAB LEF + 1: PRINT "DO";
1427  POKE ZR,P2 - 3 * B2: POKE WN,BDU
    R: CALL TW
1428  RETURN
1450 OO = 1
1452  GOSUB 70
1454  CALL  - 936
1456  COLOR= 3
1457 FLAG = 1
1458  FOR JJ = 1 TO 8
1460  ON JJ GOSUB 1060,1110,1160,1210,
    1260,1310,1360,1410
1462  NEXT JJ
1464  FOR I = 1 TO 30: NEXT I
1465  COLOR= 13
1466  FOR JJ = 8 TO 1 STEP  - 1
1468  ON JJ GOSUB 1060,1110,1160,1210,
    1260,1310,1360,1410
1470  NEXT JJ
1471 FLAG = 0
1472  GOTO 100
1500  GOTO 2252
1550  GOSUB 8100
1554 R =  FN RN(3) + 1
1555 R2 =  FN RN(4) + 1
1557 S = R + R2: IF S > 6 THEN 1554
1558  VTAB 24: PRINT R;"      +      "
    ;R2;"     =      ";
1560 TP = R
1561 SDE = 0
1562  GOSUB 1575
1563  FOR K = 1 TO 100: NEXT K
1564  GOSUB 1580
1565  FOR K = 1 TO 100: NEXT K
1566 TP = R2:SDE = 2 * (R + 1) + 6
1568  GOSUB 1575
1569 SDE = SDE + 2 * (R2 + 1)
1570  GOSUB 1585: REM  - =
1571  FOR J = 1 TO 300: NEXT J
1572 TP = S:SDE = SDE + 6
1573  GOSUB 1575
1574  PRINT S: GOTO 100
1575  FOR I = SDE TO SDE + 2 * (TP - 1
    ) STEP 2
1576  VLIN 15,25 AT I: POKE ZR,200 - I
    : POKE WN,50: CALL TW
1577  FOR K = 1 TO 100: NEXT K
1578  NEXT I
1579  RETURN
1580  HLIN 2 * R + 1,2 * R + 5 AT 20
1581  POKE ZR,20: POKE WN,20: CALL TW
1582  VLIN 18,22 AT 2 * R + 3
1583  RETURN
1585  HLIN SDE,SDE + 4 AT 19: POKE ZR,
    20: POKE WN,20: CALL TW
1586  HLIN SDE,SDE + 4 AT 21: POKE ZR,
    20: POKE WN,20: CALL TW
1588  RETURN
1600  GR
1602 L =  FN RN(4) + 2
1604 B =  FN RN(3) + 6
1606 WT = 300
1610 B1 = B:L1 = L
1612  ON L1 GOSUB 2004,1955,2854,4210,
    4602,4104,4001,4810,5110
1614  FOR J = 1 TO WT: NEXT J
1616  GR : CALL  - 936
1618  COLOR= 1
1620  FOR I = 1 TO 6
1622  PLOT 17 + I,20 - I
1624  PLOT 17 + I,20 + I
1626  NEXT I
1628  PLOT 17,20
1630  VTAB 23: HTAB 12: PRINT "IS SMAL
    LER THAN"
1632  FOR J = 1 TO WT: NEXT J
1634  GR : CALL  - 936
1636  ON B1 GOSUB 2004,1955,2854,4210,
    4602,4104,4001,4810,5110
1640  GOTO 100
1650  GOTO 850
1700  GR
1702 L =  FN RN(4) + 2
1704 B =  FN RN(3) + 6
1706 WT = 300
1710 B1 = B:L1 = L
1712  ON B1 GOSUB 2004,1955,2854,4210,
    4602,4104,4001,4810,5110
1715  FOR J = 1 TO WT: NEXT J
1717  GR : CALL  - 936
1718  COLOR= 1
1720  FOR I = 1 TO 6
1722  PLOT 21 - I,20 - I
1724  PLOT 21 - I,20 + I
1726  NEXT I
1730  PLOT 21,20
1731  VTAB 23: HTAB 12: PRINT "IS BIGG
    ER THAN"
1732  FOR I = 1 TO WT: NEXT I
1734  GR : CALL  - 936
1740  ON L1 GOSUB 2004,1955,2854,4210,
    4602,4104,4001,4810,5110
1749  GOTO 100
1750  GOSUB 6460
1756  GOTO 100
1800  GOTO 4200
1850  GOTO 951
1900  GOTO 2755
1901  COLOR= 0: FOR I = 15 TO 25: HLIN
    15,25 AT I: NEXT I
1902  COLOR= 13
1904  PLOT 20,20
1906  FOR I = 1 TO 300: NEXT I
1907  GOSUB 1926
1908 NUM =  FN RN(10) + 10
1910  FOR K = 1 TO NUM
1912  GOSUB 8100
1914 X =  FN RN(40):Y =  FN RN(40)
1915  IF X > 14 AND X < 26 AND Y > 15 AND
    Y < 26 THEN 1914
1916  PLOT X,Y
1918  POKE ZR,40 + X + Y: POKE WN,10: CALL
    TW
1920  NEXT K
1921  GOSUB 2780: IF CUT = 1 THEN 100
1922  GOSUB 310
1924  GOTO 100
1926  COLOR= 0: PLOT 20,20:X = 20:Y =
    20
1928  COLOR= 13
1930  FOR I = 1 TO 5
1932  PLOT X + I,Y - I: PLOT X + I,Y: PLOT
    X + I,Y + I: PLOT X,Y + I
1934  PLOT X - I,Y + I: PLOT X - I,Y: PLOT
    X - I,Y - I: PLOT X,Y - I
1936  FOR J = 1 TO 50: NEXT J
1938  NEXT I
1940  RETURN
1950  GOSUB 1955
1954  GOTO 100
1955 X =  FN RN(20) + 3:Y =  FN RN(23)
    + 2
1956  COLOR= 8
1960  HLIN X,X + 1 AT Y: HLIN X + 13,X
    + 14 AT Y
1962  HLIN X,X + 3 AT Y + 1: HLIN X +
    11,X + 14 AT Y + 1
1964  HLIN X,X + 4 AT Y + 2: HLIN X +
    10,X + 14 AT Y + 2
1966  HLIN X,X + 4 AT Y + 3: HLIN X +
    10,X + 14 AT Y + 3
1968  COLOR= 9: HLIN X + 5,X + 9 AT Y +
    3: COLOR= 8
1970  HLIN X,X + 2 AT Y + 4: HLIN X +
    12,X + 14 AT Y + 4
1972  COLOR= 9: HLIN X + 3,X + 11 AT Y
    + 4
1974  FOR I = 5 TO 11
1976  HLIN X,X + 14 AT Y + I
1978  NEXT I
1980  COLOR= 15: HLIN X + 3,X + 6 AT Y
    + 6: HLIN X + 10,X + 12 AT Y + 6
1982  PLOT X + 2,Y + 7: PLOT X + 6,Y +
    7: PLOT X + 8,Y + 7: PLOT X + 12,Y
    + 7
1984  FOR I = 8 TO 12: HLIN X + 1,X +
    13 AT Y + I: NEXT I
1986  HLIN X + 3,X + 11 AT Y + 13
1988  HLIN X + 6,X + 8 AT Y + 14
1990  COLOR= 4: FOR I = 7 TO 9: HLIN X
    + 3,X + 5 AT Y + I: HLIN X + 10,X
    + 12 AT Y + I: NEXT I
1992  COLOR= 13: VLIN Y + 8,Y + 9 AT X
    + 4: VLIN Y + 8,Y + 9 AT X + 11
1994  COLOR= 11: HLIN X + 6,X + 8 AT Y
    + 10: PLOT X + 7,Y + 11
1996  COLOR= 0: HLIN X + 3,X + 5 AT Y +
    11: HLIN X + 9,X + 11 AT Y + 11
1997  COLOR= 11: HLIN X + 6,X + 8 AT Y
    + 13
1998  VTAB 23: HTAB 20: PRINT "CAT"
1999  RETURN
2000  GR
2002 X = 26:Y = 5: COLOR= 8
2003  GOSUB 2004: GOTO 2029
2004  FOR I = 0 TO 4
2006  HLIN X - 5 - I,X + 5 + I AT Y +
    I
2008  NEXT I
2010  FOR I = 5 TO 8: HLIN X - 5,X + 5
    AT Y + I: NEXT I
2012 X1 = X - 6:X2 = X + 6: HLIN X1,X2
    AT Y + 9: HLIN X1,X2 AT Y + 14
2014  FOR I = 10 TO 13: HLIN X - 7,X +
    7 AT Y + I: NEXT I
2016  HLIN X - 5,X + 5 AT Y + 15
2018  HLIN X - 10,X - 7 AT Y + 5: HLIN
    X + 7,X + 10 AT Y + 5
2020  FOR I = 6 TO 7: HLIN X - 10,X -
    8 AT Y + I: HLIN X + 8,X + 10 AT Y
    + I: NEXT I
2021  COLOR= 15: FOR I = 6 TO 7: HLIN
    X - 4,X - 2 AT Y + I: HLIN X + 2,X
    + 4 AT Y + I: NEXT I
2022  COLOR= 0: HLIN X - 4,X - 2 AT Y +
    4: HLIN X + 2,X + 4 AT Y + 4
2024  VLIN Y + 9,Y + 12 AT X: VLIN Y +
    9,Y + 10 AT X - 1: VLIN Y + 9,Y +
    10 AT X + 1
2025  HLIN X - 3,X - 1 AT Y + 13: HLIN
    X + 1,X + 3 AT Y + 13: PLOT X - 4,
    Y + 12: PLOT X + 4,Y + 12
```

```
2026  PLOT X - 3,Y + 6: PLOT X + 3,Y +
    6
2028  RETURN
2029  FOR I = 1 TO 300: NEXT I: COLOR=
    9
2030  FOR I = 25 TO 37: HLIN 0,15 AT I
    : NEXT I: VLIN 23,25 AT 0: VLIN 23
    ,25 AT 15
2031  HLIN 20,22 AT 35
2032  COLOR= 15: HLIN 17,18 AT 37
2033  COLOR= 13: HLIN 0,15 AT 30: HLIN
    20,22 AT 33: HLIN 20,22 AT 34
2034  HTAB 7: PRINT "TUB      SOAP BRU
    SH"
2035  FOR I = 1 TO 200: NEXT I
2036  COLOR= 15: PLOT X - 3,Y + 6: PLOT
    X + 3,Y + 6
2037  COLOR= 0: PLOT X - 4,Y + 7: PLOT
    X + 2,Y + 7
2038  FOR I = 1 TO 300: NEXT I
2040  COLOR= 8: PLOT X - 4,Y + 12: PLOT
    X + 4,Y + 12
2041  COLOR= 0: PLOT X - 4,Y + 14: PLOT
    X + 4,Y + 14
2042  FOR W = 1 TO 300: NEXT W
2043  COLOR= 0: GOSUB 2004
2045  GOSUB 2780: IF CUT = 1 THEN 100
2046  GOSUB 338
2047  GR : CALL  - 936
2048  GOTO 100
2050  COLOR= 6
2058  HLIN 16,24 AT 30: VLIN 15,30 AT
    16: VLIN 15,30 AT 24
2060  VTAB 24: HTAB 19: PRINT "EMPTY"
2062  FOR I = 1 TO 200: NEXT I: COLOR=
    15
2067  FOR J = 29 TO 15 STEP  - 1
2070  FOR I = 17 TO 23
2072  PLOT I,J
2074  NEXT I: FOR K = 1 TO 100: NEXT K

2075  NEXT J
2080  CALL  - 936: VTAB 24: HTAB 19: PRINT
    "FULL": GOTO 100
2100  GOSUB 5600
2104  FOR I = 1 TO 13:DURV(I) = BDUR: NEXT
    I
2105 DURV(7) = DURV(7) + 2 * DRINC
2106 DURV(12) = DURV(12) + 3 * DRINC
2108 TN(1) = 4:TN(2) = 4:TN(3) = 4:TN(
    4) = 1
2110 TN(5) = 2:TN(6) = 2:TN(7) = 1:TN(
    8) = 6
2112 TN(9) = 6:TN(10) = 5:TN(11) = 5
2114 TN(12) = 4:TN(13) = 1
2115 XLN = 13
2116  GOSUB 82
2117 XLN = 6: GOSUB 82: GR
2118 X =  FN RN(20) + 10:Y =  FN RN(15
    ) + 10
2119  GOSUB 4360
2120 RX =  FN RN(WL) + 1: IF USED(RX) =
    1 THEN 2120
2121  GOTO 2142
2122  FOR I = 1 TO 200: NEXT I
2125  FOR I = 1 TO 5:TN(I) = TN(I + 7)
    : NEXT I
2127 DURV(5) = DURV(7)
2128 XLN = 5: GOSUB 82
2130  GOTO 100
2141  COLOR= 8: GOSUB 2004: VTAB 23: HTAB
    20: PRINT "DOG": GOTO 2122
2142 USED(RX) = 1
2143  IF RX = 1 THEN 2141
2144 WX = WR(RX)
2146  ON RX GOSUB 2004,1955,2854,4210,
    4602,4104,4001,4810,5110
2147  GOTO 2122
2150  GOTO 5100
2200  GOTO 4800
2202 R =  FN RN(3) + 1
2204  ON R GOSUB 2208,2223,2238
2206  GOTO 100
2208  REM   - STRAW HAT
2209  GOSUB 2211: GOSUB 2780: IF CUT =
    1 THEN  RETURN
2210  GOSUB 260: RETURN
2211 X =  FN RN(20) + 5:Y =  FN RN(30)

2212  FOR I = X TO X + 9: FOR J = Y TO
    Y + 3
2214  IF  INT ((I + J) / 2) * 2 = (I +
    J) THEN 2217
2215  COLOR= 13: PLOT I,J: GOTO 2218
2217  COLOR= 15: PLOT I,J
2218  NEXT J: NEXT I
2219  COLOR= 1: HLIN X,X + 9 AT Y + 4
2220  FOR I = X - 5 TO X + 14 STEP 2
2221  COLOR= 13: PLOT I,Y + 5: COLOR=
    15: PLOT I + 1,Y + 5
2222  NEXT I: RETURN
2223  REM   - TOP HAT
2224  FOR I = 0 TO 30 STEP 10: FOR J =
    0 TO 9
2225  IF  INT ( INT (I / 10) / 2) * 2 =
     INT (I / 10) THEN 2228
2226  COLOR= 15: HLIN 0,39 AT I + J: GOTO
    2229
2228  COLOR= 1: HLIN 0,39 AT I + J
2229  NEXT J
2230  NEXT I
2232  COLOR= 2: HLIN 10,24 AT 25
2234  FOR J = 15 TO 24: VLIN 8,24 AT J
    : NEXT J
2235  COLOR= 13: HLIN 15,24 AT 24
2236  VLIN 6,11 AT 15: GOSUB 352
2237  RETURN
2238  REM   - BASEBALL CAP
2240 X =  FN RN(15):Y =  FN RN(25) + 1
    5
2242  COLOR= 13
2244  FOR I = 0 TO 2: VLIN Y - 2 * (I +
    1),Y AT X + I: VLIN Y - (I + 6),Y AT
    X + 3 + I: NEXT I
2246  VLIN Y - 9,Y - 8 AT X + 6: VLIN
    Y - 9,Y - 8 AT X + 9: VLIN Y - 8,Y
    - 7 AT X + 10
2248  FOR I = 0 TO 1: VLIN Y - 8 + 2 *
    I,Y AT X + 11 + I: NEXT I
2249  GOSUB 2452: GOTO 100
2250  GOTO 5250: REM   - ICE CREAM
2252 X =  FN RN(35):Y = 35:L =  - 20 -
     FN RN(15):D = 20:SP =  -  RND (5)
    - 1
2254  GOSUB 40
2256  GOTO 100
2300  GOTO 2510
2302  IF P = PL THEN 2312
2304  GR : COLOR= 13
2305  POKE WN,5
2306  HLIN 0,39 AT 30
2308  VTAB 21
2309  PRINT "  0   1   2   3   4   5
     6   7   8   9"
2310 JC = 1:JL = 1: GOTO 2315
2312  COLOR= 0: FOR KX = 1 TO 9: HLIN
    4 * JL,4 * JL + 3 AT 30 - KX: NEXT
    KX
2313 JC = JC + 1
2314 JL = JC -  INT (JC / 10) * 10
2315  IF JL = 0 THEN 2320
2316  GOTO 8095
2317  PLOT 4 * JL + 3,30 - KX
2318  POKE ZR,(250 - 5 * KX): POKE WN,
    10: CALL TW
2319  NEXT KX
2320 JH =  INT (JC / 10)
2322  IF JH < 1 THEN 2345
2324  IF JH > 9 THEN 2330
2325  VTAB 21: HTAB 4 * JL + 3: PRINT
    JH: GOSUB 2326: GOTO 2345
2326  FOR MX = 1 TO JH: GOTO 8120
2327  PLOT 4 * JL + 2,30 - MX: POKE ZR
    ,170 - 5 * MX: POKE WN,10: CALL TW

2328  NEXT MX
2329  RETURN
2330 UP =  INT (JC / 100):JH = JC -  INT
    (JC / 10) * 10
2331  IF UP > 9 THEN 2340
2332  VTAB 21: HTAB 4 * JL + 2: PRINT
    UP;JH
2333  FOR NX = 1 TO UP: GOTO 8130
2334  PLOT 4 * JL + 1,30 - NX: POKE ZR
    ,90 - 5 * NX: POKE WN,10: CALL TW
2335  NEXT NX
2336  IF JH = 0 THEN 2345
2337  GOSUB 2326: GOTO 2345
2340  VTAB 23: PRINT JC;"  TOO BIG!"
2341  GR : COLOR= 13
2342  GOTO 2306
2345 FGJ = 1:PL = 0
2347  GOTO 100
2350  GOTO 5800
2352  GOTO 100
2400  GOSUB 4600
2449  GOTO 100
2450  GOTO 2470
2452  REM   - CONTINUE B.B. CAP
2454  VLIN Y - 4,Y AT X + 13: VLIN Y -
    3,Y AT X + 14
2455  PLOT X + 10,Y - 6
2456  COLOR= 4
2458  FOR I = 0 TO 2
2460  VLIN Y - 9 + 2 * (2 - I),Y AT X +
    5 + I
2462  VLIN Y - 9 + 2 * I,Y AT X + 8 +
    I
2463  NEXT I
2464  HLIN X,X + 22 AT Y
2465  GOSUB 2780: IF CUT = 1 THEN  RETURN

2466  GOSUB 272
2468  RETURN
2470  GR : COLOR= 3
2472 X = 19:Y = 5
2474  FOR R = 0 TO 4
2476  HLIN X - 5,X + 5 AT Y + R
2478  NEXT R
2480  FOR I = 3 TO 5
2481  VLIN Y,Y + 23 AT X - I
2482  VLIN Y,Y + 23 AT X + I
2483  NEXT I
```

```
2485  COLOR= 1
2486  FOR I = 3 TO 5
2487  VLIN Y + 24,Y + 30 AT X - I
2488  VLIN Y + 24,Y + 30 AT X + I
2489  NEXT I
2490  VTAB 23: HTAB 15: PRINT "MAGNET"

2491  FOR W = 1 TO 200: NEXT W:WT = 55

2492 NM =  FN RN(15) + 10
2493  FOR I = 1 TO NM:WT = WT - 2
2494  COLOR=  FN RN(13) + 3:XP =  FN R
    N(10) + 1:YP =  FN RN(17) + Y + 15

2495  FOR J = 1 TO WT: NEXT J: IF XP <
    6 THEN XZ = X - 1 - XP
2496  IF XP > 5 THEN XZ = X - 4 + XP
2497  PLOT XZ,YP: POKE ZR,100 + XP + Y
    P: POKE WN,15: CALL TW
2498  NEXT I
2499  GOTO 100
2500  GOTO 2302
2502 X =  FN RN(35):Y = 5:L = 20 +  FN
    RN(15):D = 20:SP =  FN RN(5) + 1
2504  GOSUB 40
2506  GOTO 100
2510  REM  - JUMP
2512 C =  FN RN(15) + 1: COLOR= C
2513 X = 30:Y = 30: GOSUB 6250: GOSUB
    2540
2514  COLOR= 0: GOSUB 6250
2515 X = X - 3:Y = Y - 6
2516  IF Y < 0 THEN 2521
2517  COLOR= C: GOSUB 6250: GOSUB 2540

2518  GOTO 2514
2521 Y = 6: COLOR= C: GOSUB 6250: GOSUB
    2540
2523  COLOR= 0: GOSUB 6250
2525 X = X - 3:Y = Y + 6
2526  IF X < 0 THEN 2532
2528  COLOR= C: GOSUB 6250: GOSUB 2540

2530  GOTO 2523
2532  FOR J = 1 TO 200: NEXT J
2534 X = X + 3:Y = 30: COLOR= C: GOSUB
    6250: GOSUB 2544
2536 X = X + 5: GOSUB 6360: GOSUB 2544

2537 X = X + 5: GOSUB 6280: GOSUB 2544

2538 X = X + 6: GOSUB 6310: GOSUB 2544

2539  GOTO 100
2540  POKE ZR,Y + 50: POKE WN,20: CALL
    TW
2542  RETURN
2544  POKE ZR,2 * X + 150: POKE WN,20:
    CALL TW
2546  RETURN
2550  GOSUB 2208
2552  GOTO 100
2600  GOTO 2652
2650  GOTO 5360
2652 X = 20:Y = 15: COLOR= 11: GOSUB 2
    654: GOTO 2660
2654  HLIN X - 12,X + 12 AT Y + 12
2656  FOR I = 0 TO 12
2658  HLIN X - I,X + I AT Y + I
2659  NEXT I: RETURN
2660 XS = X:YS = Y:LAST = 39:D = 10:SP
    = 1
2661  VTAB 23: HTAB 18: PRINT "PRISM"
2662  COLOR= 15:X = 0:Y = YS + 2:L = X
    S - 3: GOSUB 30: FOR I = 1 TO 100:
    NEXT I
2663  COLOR= 1:X = XS + 1:Y = YS + 1:L
    = LAST - X: GOSUB 30
2664  GOSUB 2780: IF CUT = 1 THEN 100
2665  COLOR= 9:X = XS + 3:Y = YS + 3:L
    = LAST - X: GOSUB 30
2666  COLOR= 13:X = XS + 5:Y = YS + 5:
    L = LAST - X: GOSUB 30
2668  COLOR= 4:X = XS + 7:Y = YS + 7:L
    = LAST - X: GOSUB 30
2670  COLOR= 2:X = XS + 9:Y = YS + 9:L
    = LAST - X: GOSUB 30
2672  COLOR= 3:X = XS + 11:Y = YS + 11
    :L = LAST - X: GOSUB 30
2698  GOTO 100
2700  GOTO 2851
2750  GOTO 1901
2752 OO = 1
2755  GOSUB 4000
2757  COLOR= 9: PLOT X - 4,Y + 24: PLOT
    X + 4,Y + 24
2758  COLOR= 8: HLIN X - 4,X + 4 AT Y +
    8: HLIN X - 4,X + 4 AT Y + 9
2760  FOR JJ = 1 TO 200: NEXT JJ
2767  GOSUB 2789
2770  COLOR= 15: HLIN X - 4,X - 2 AT Y
    + 8: HLIN X + 2,X + 4 AT Y + 8
2772  HLIN X - 4,X - 2 AT Y + 9: HLIN
    X + 2,X + 4 AT Y + 9
2778  GOTO 100
2779 OO = 1
2780 P3 =  PEEK ( - 16384)
2781  POKE  - 16368,0
2782  IF P3 = 155 OR P3 = 141 THEN CUT
    = 1
2784  RETURN
2789  COLOR= 1: HLIN X - 3,X + 3 AT Y +
    25
2790  HLIN X - 3,X + 3 AT Y + 26: PLOT
    X - 4,Y + 24: PLOT X + 4,Y + 24
2791  RETURN
2800 OO = 1
2802  GOSUB 2238
2804  GOTO 100
2850  GOTO 5700
2851  GOSUB 2854
2852  GOTO 100
2854 X =  FN RN(19) + 3:Y =  FN RN(22)

2855  COLOR= 15
2858  HLIN X,X + 2 AT Y: HLIN X + 12,X
    + 14 AT Y
2860  HLIN X,X + 3 AT Y + 1: HLIN X +
    11,X + 14 AT Y + 1
2862  HLIN X,X + 4 AT Y + 2: HLIN X +
    10,X + 14 AT Y + 2
2864  HLIN X,X + 5 AT Y + 3: HLIN X +
    9,X + 14 AT Y + 3
2866  FOR I = 4 TO 5: HLIN X + 1,X + 5
    AT Y + I: HLIN X + 9,X + 13 AT Y +
    I: NEXT I
2868  HLIN X + 2,X + 5 AT Y + 6: HLIN
    X + 9,X + 12 AT Y + 6
2870  HLIN X + 3,X + 11 AT Y + 7
2872  FOR I = 8 TO 9: HLIN X + 2,X + 1
    2 AT Y + I: NEXT I
2874  FOR I = 10 TO 14: HLIN X + 1,X +
    13 AT Y + I: NEXT I
2876  HLIN X + 2,X + 12 AT Y + 15
2877  HLIN X + 4,X + 10 AT Y + 16
2878  HLIN X + 6,X + 8 AT Y + 17
2880  COLOR= 11: PLOT X + 1,Y + 2: PLOT
    X + 13,Y + 2
2882  VLIN Y + 2,Y + 3 AT X + 2: VLIN
    Y + 2,Y + 3 AT X + 12
2884  VLIN Y + 3,Y + 5 AT X + 3: VLIN
    Y + 3,Y + 5 AT X + 11
2886  VLIN Y + 5,Y + 7 AT X + 4: VLIN
    Y + 5,Y + 7 AT X + 10
2888  HLIN X + 6,X + 8 AT Y + 13: PLOT
    X + 7,Y + 14: HLIN X + 6,X + 8 AT
    Y + 16
2890  COLOR= 2: FOR I = 10 TO 11: HLIN
    X + 4,X + 5 AT Y + I: HLIN X + 9,X
    + 10 AT Y + I: NEXT I
2892  COLOR= 0: HLIN X + 3,X + 5 AT Y +
    14: HLIN X + 9,X + 11 AT Y + 14
2894  VTAB 23: HTAB 20: PRINT "RABBIT"

2896  RETURN
2900  GR : VTAB 24: HTAB 15: PRINT "VO
    LCANO"
2904  COLOR= 4: FOR I = 30 TO 39: HLIN
    0,39 AT I: NEXT I
2906 X = 19:Y = 23: COLOR= 10: GOSUB 2
    654: FOR J = 1 TO 200: NEXT J
2910  COLOR= 1: FOR I = 0 TO 5: PLOT X
    ,Y - I: GOSUB 2940: NEXT I
2912  FOR I = 4 TO 0 STEP  - 1: PLOT X
    + 1,Y - I: PLOT X - 1,Y - I: FOR
    J = 1 TO 50: NEXT J: NEXT I
2915  FOR I = 1 TO 6: COLOR= 0: HLIN X
    - 1,X + 1 AT Y - 6 + I
2917  COLOR= 1: HLIN X - 2 + I,X + I AT
    Y + I: GOSUB 2942
2919  NEXT I
2920  FOR J = 1 TO 100: NEXT J
2922  FOR I = 1 TO 4: COLOR= 0: HLIN X
    - 3 + I,X + I AT Y + I
2924  COLOR= 1: HLIN X + 4 + I,X + I +
    6 AT Y + 6 + I: GOSUB 2942
2925  NEXT I
2927  FOR I = 9 TO 11: HLIN X + I,X +
    I + 2 AT Y + I + 2: GOSUB 2942: NEXT
    I
2930  GOTO 100
2940  FOR J = 1 TO 100 - 5 * I: NEXT J
    : RETURN
2942  FOR J = 1 TO 50: NEXT J: RETURN

2950  GOTO 5500
3000 OO = 1
3004 C =  FN RN(15) + 1: COLOR= C
3006  FOR I = 0 TO 19
3008  PLOT I,I: PLOT I,38 - I
3010  PLOT 38 - I,I: PLOT 38 - I,38 -
    I
3012  NEXT I
3015  FOR SI = 0 TO 19
3017  COLOR= C
3018  HLIN 19 - SI,19 + SI AT 19 - SI
3019  HLIN 19 - SI,19 + SI AT 19 + SI
3020  VLIN 19 - SI,19 + SI AT 19 - SI
3021  VLIN 19 - SI,19 + SI AT 19 + SI
3023  COLOR= 0
3025  PLOT 19 - SI,19 - SI: PLOT 19 +
    SI,19 - SI
3026  PLOT 19 + SI,19 + SI: PLOT 19 -
```

```
   SI,19 + SI
3028  NEXT SI
3040  GOTO 100
3050  GOSUB 2223
3052  GOTO 100
3100  GOSUB 8100
3106 X1 = 10 +  FN RN(10)
3107 YO = 0
3108  FOR I = X1 TO 20
3110  PLOT I,YO
3111  POKE ZR,YO + I: POKE WN,20: CALL
    TU
3112 YO = YO + 1
3114  NEXT I
3115  IF YO + 20 - X1 > 39 THEN 3147
3116  GOSUB 8100
3118  FOR I = 20 TO X1 STEP  - 1
3120  PLOT I,YO
3122  POKE ZR,YO + I: POKE WN,20: CALL
    TU
3124 YO = YO + 1
3126  NEXT I
3127  IF YO + 20 - X1 > 39 THEN 3147
3128  GOSUB 8100
3130  GOTO 3108
3147  VTAB 23: HTAB 15: PRINT "ZIG-ZAG
    "
3149  GOTO 100
3150  GOTO 2200
3200  TEXT : CALL  - 936
3202  GOTO 9999
4000  GOSUB 4001: GOTO 100
4001  GOSUB 4005: GOSUB 4002: RETURN
4002  FOR J = 1 TO 150: NEXT J: COLOR=
    15
4003  HLIN X - 4,X - 2 AT Y + 8: HLIN
    X + 2,X + 4 AT Y + 8: HLIN X - 4,X
     - 2 AT Y + 9: HLIN X + 2,X + 4 AT
    Y + 9
4004  COLOR= 1: PLOT X - 4,Y + 24: PLOT
    X + 4,Y + 24: RETURN
4005  COLOR= 8
4010 X =  FN RN(14) + 13:Y = 7
4012  FOR J = 0 TO 7
4014  HLIN X - 12,X + 12 AT Y + J
4015  NEXT J
4017  FOR J = 1 TO 3
4019  HLIN X - 11,X - 3 AT Y - J: HLIN
    X + 3,X + 11 AT Y - J
4020  NEXT J
4022  HLIN X - 10,X - 4 AT Y - 4: HLIN
    X + 4,X + 10 AT Y - 4
4024  HLIN X - 9,X - 5 AT Y - 5: HLIN
    X + 5,X + 9 AT Y - 5
4026  HLIN X - 11,X + 11 AT Y + 8
4028  HLIN X - 10,X + 10 AT Y + 9: HLIN
    X - 10,X + 10 AT Y + 10
4030  HLIN X - 8,X + 8 AT Y + 11
4032  HLIN X - 9,X + 9 AT Y + 12
4034  HLIN X - 10,X + 10 AT Y + 13
4036  HLIN X - 10,X + 10 AT Y + 14
4038  HLIN X - 11,X + 11 AT Y + 15
4040  HLIN X - 11,X + 11 AT Y + 16
4042  FOR J = 17 TO 24
4044  HLIN X - 12,X + 12 AT Y + J
4045  NEXT J
4046  HLIN X - 11,X + 11 AT Y + 25
4048  HLIN X - 10,X + 10 AT Y + 26
4050  HLIN X - 9,X + 9 AT Y + 27
4052  HLIN X - 9,X + 9 AT Y + 28
4054  HLIN X - 8,X + 8 AT Y + 29
4056  HLIN X - 7,X + 7 AT Y + 30
4058  HLIN X - 5,X + 5 AT Y + 31
4060  COLOR= 15
4066  FOR J = 10 TO 13
4068  HLIN X - 5,X - 2 AT Y + J: HLIN
    X + 2,X + 5 AT Y + J
4070  NEXT J
4072  COLOR= 0
4074  PLOT X - 3,Y + 10: PLOT X - 3,Y +
    11
4075  PLOT X + 3,Y + 10: PLOT X + 3,Y +
    11
4076  COLOR= 9
4078  FOR J = 17 TO 27
4080  HLIN X - 5,X + 5 AT Y + J
4082  NEXT J
4084  COLOR= 0
4086  FOR J = 18 TO 22
4088  HLIN X - 2,X + 2 AT Y + J
4090  NEXT J
4091  COLOR= 9: PLOT X - 2,Y + 22: PLOT
    X + 2,Y + 22
4092  COLOR= 1
4094  HLIN X - 3,X + 3 AT Y + 25
4095  HLIN X - 3,X + 3 AT Y + 26
4098  VTAB 23: HTAB 18: PRINT "BEAR": RETURN

4100  GOSUB 4104: GOTO 100
4104 X =  FN RN(12) + 13:Y =  FN RN(5)
     + 8
4106  COLOR= 9
4108  FOR I = 9 TO 12
4109  HLIN X - 7,X + 7 AT Y + I
4110  NEXT I
4112  HLIN X - 6,X - 3 AT Y + 8: HLIN
    X + 3,X + 6 AT Y + 8
4113  HLIN X - 7,X - 4 AT Y + 7: HLIN
    X + 4,X + 7 AT Y + 7
4114  HLIN X - 7,X - 5 AT Y + 6: HLIN
    X + 5,X + 7 AT Y + 6
4115  HLIN X - 7,X - 6 AT Y + 5: HLIN
    X + 6,X + 7 AT Y + 5
4116  HLIN X - 11,X - 6 AT Y + 4: HLIN
    X + 6,X + 11 AT Y + 4
4117  VLIN Y + 5,Y + 6 AT X - 12: VLIN
    Y + 5,Y + 6 AT X + 12
4118  HLIN X - 11,X - 8 AT Y + 8: HLIN
    X + 8,X + 11 AT Y + 8
4120  HLIN X - 6,X + 6 AT Y + 13
4122  HLIN X - 5,X + 5 AT Y + 14
4123  HLIN X - 5,X + 5 AT Y + 15
4124  HLIN X - 4,X + 4 AT Y + 16
4125  HLIN X - 4,X + 4 AT Y + 17
4126  HLIN X - 3,X + 3 AT Y + 18
4127  HLIN X - 3,X + 3 AT Y + 19
4128  HLIN X - 2,X + 2 AT Y + 20
4129  HLIN X - 2,X + 2 AT Y + 21
4130  HLIN X - 2,X + 2 AT Y + 22
4131  PLOT X,Y + 23
4132  COLOR= 15
4133  HLIN X - 5,X + 5 AT Y + 4
4134  FOR I = 1 TO 4
4135  HLIN X - 6 + I,X + 6 - I AT Y +
    4 + I
4136  HLIN X - 6 + I,X + 6 - I AT Y +
    4 - I
4137  NEXT I
4138  COLOR= 11
4139  HLIN X - 11,X - 8 AT Y + 5: HLIN
    X + 8,X + 11 AT Y + 5
4140  HLIN X - 11,X - 8 AT Y + 6: HLIN
    X + 8,X + 11 AT Y + 6
4141  HLIN X - 11,X - 8 AT Y + 7: HLIN
    X + 8,X + 11 AT Y + 7
4142  HLIN X - 2,X - 1 AT Y + 23: HLIN
    X + 1,X + 2 AT Y + 23
4143  HLIN X - 3,X + 3 AT Y + 24
4144  HLIN X - 3,X + 3 AT Y + 25
4145  HLIN X - 3,X + 3 AT Y + 26
4146  HLIN X - 2,X + 2 AT Y + 27
4147  COLOR= 15
4148  VLIN Y + 11,Y + 13 AT X - 4
4149  VLIN Y + 11,Y + 13 AT X - 3
4150  VLIN Y + 11,Y + 13 AT X + 3
4151  VLIN Y + 11,Y + 13 AT X + 4
4152  COLOR= 0
4153  PLOT X - 4,Y + 12: PLOT X + 4,Y +
    12
4154  PLOT X - 1,Y + 25: PLOT X + 1,Y +
    25
4160  COLOR= 10
4162  VLIN Y - 4,Y + 1 AT X - 8
4163  VLIN Y - 4,Y + 1 AT X + 8
4165  VLIN Y - 5,Y + 3 AT X - 7
4166  VLIN Y - 5,Y + 3 AT X + 7
4168  VLIN Y - 6,Y + 3 AT X - 6
4169  VLIN Y - 6,Y + 3 AT X + 6
4171  VLIN Y - 7,Y + 2 AT X - 5
4172  VLIN Y - 7,Y + 2 AT X + 5
4174  VLIN Y - 7,Y - 4 AT X - 4
4175  VLIN Y - 7,Y - 4 AT X + 4
4177  VLIN Y - 1,Y + 1 AT X - 4
4178  VLIN Y - 1,Y + 1 AT X + 4
4180  VLIN Y - 8,Y - 6 AT X - 3
4181  VLIN Y - 8,Y - 6 AT X + 3
4183  HLIN X - 2,X - 1 AT Y - 8
4184  HLIN X + 1,X + 2 AT Y - 8
4186  VTAB 23: HTAB 20: PRINT "COW"
4190  RETURN
4200  GOSUB 4210: GOTO 100
4210 X =  FN RN(6) + 15:Y =  FN RN(8) +
    5
4212  COLOR= 11
4214  HLIN X - 3,X + 3 AT Y
4216  HLIN X - 5,X + 5 AT Y + 1
4218  HLIN X - 12,X + 12 AT Y + 2
4220  HLIN X - 12,X + 12 AT Y + 3
4222  HLIN X - 12,X + 12 AT Y + 4
4224  HLIN X - 12,X + 12 AT Y + 5
4226  HLIN X - 12,X + 12 AT Y + 6
4228  HLIN X - 10,X + 10 AT Y + 7
4230  HLIN X - 6,X + 6 AT Y + 8
4232  HLIN X - 6,X + 6 AT Y + 9
4236  FOR J = 10 TO 17
4238  HLIN X - 7,X + 7 AT Y + J
4240  NEXT J
4242  HLIN X - 6,X + 6 AT Y + 18
4244  HLIN X - 6,X + 6 AT Y + 19
4246  HLIN X - 5,X + 5 AT Y + 20
4248  HLIN X - 5,X + 5 AT Y + 21
4250  HLIN X - 4,X + 4 AT Y + 22
4252  HLIN X - 2,X + 2 AT Y + 23
4253  HLIN X - 1,X + 1 AT Y + 24
4254  HLIN X - 11,X - 8 AT Y + 1
4256  HLIN X + 8,X + 11 AT Y + 1
4262  COLOR= 0
4264  PLOT X - 4,Y + 9: PLOT X + 4,Y +
    9
4270  COLOR= 10
4280  FOR J = 13 TO 19
4282  HLIN X - 3,X + 3 AT Y + J
4284  NEXT J
```

```
4290  HLIN X - 1,X + 1 AT Y + 20
4293  VLIN Y + 11,Y + 14 AT X - 2
4294  VLIN Y + 11,Y + 14 AT X + 2
4295  PLOT X - 3,Y + 10: PLOT X + 3,Y +
    10
4276  COLOR= 1
4300  HLIN X - 1,X + 1 AT Y + 21
4302  PLOT X,Y + 22
4304  COLOR= 8
4306  FOR J = 15 TO 17
4308  HLIN X - 2,X - 1 AT Y + J
4310  HLIN X + 1,X + 2 AT Y + J
4312  NEXT J
4314  VTAB 23: HTAB 20: PRINT "PIG"
4316  FOR I = 1 TO 200: NEXT I
4318  COLOR= 1: PLOT X - 2,Y + 20: PLOT
    X + 2,Y + 20
4320  RETURN
4350 WL = 9
4352 WR(1) = 2004:WR(2) = 1955:WR(3) =
    2854
4354 WR(4) = 4210:WR(5) = 4602:WR(6) =
    4104
4356 WR(7) = 4001:WR(8) = 4810:WR(9) =
    5110
4359  RETURN
4360  FOR I = 1 TO WL
4362  IF USED(I) = 0 THEN 4372
4364  NEXT I
4365  FOR I = 1 TO WL
4367 USED(I) = 0
4370  NEXT I
4372  RETURN
4600  REM  - LAMB
4601  GOSUB 4602: GOSUB 4730: GOTO 100

4602 X =  FN RN(6) + 16:Y =  FN RN(10)
    + 2
4605  COLOR= 15
4610  HLIN X - 3,X + 3 AT Y
4612  HLIN X - 14,X - 9 AT Y - 1
4614  HLIN X + 9,X + 14 AT Y - 1
4616  HLIN X - 15,X - 7 AT Y
4618  HLIN X + 7,X + 15 AT Y
4620  HLIN X - 15,X - 7 AT Y + 1
4622  HLIN X + 7,X + 15 AT Y + 1
4624  HLIN X - 5,X + 5 AT Y + 1
4626  HLIN X - 15,X + 15 AT Y + 2
4628  HLIN X - 14,X + 14 AT Y + 3
4630  HLIN X - 14,X + 14 AT Y + 4
4632  HLIN X - 12,X + 12 AT Y + 5
4634  HLIN X - 10,X + 10 AT Y + 6
4636  FOR J = 7 TO 15
4638  HLIN X - 8,X + 8 AT Y + J
4640  NEXT J
4642  HLIN X - 7,X + 7 AT Y + 16
4644  HLIN X - 7,X + 7 AT Y + 17
4646  HLIN X - 6,X + 6 AT Y + 18
4650  COLOR= 0
4652  HLIN X - 4,X - 3 AT Y + 6
4654  HLIN X + 3,X + 4 AT Y + 6
4656  HLIN X - 4,X + 4 AT Y + 7
4658  FOR J = 8 TO 15
4660  HLIN X - 5,X + 5 AT Y + J
4662  NEXT J
4664  FOR J = 16 TO 18
4666  HLIN X - 4,X + 4 AT Y + J
4667  NEXT J
4668  HLIN X - 3,X + 3 AT Y + 19
4670  HLIN X - 2,X + 2 AT Y + 20
4676  COLOR= 6
4678  VLIN Y + 9,Y + 12 AT X - 4
4680  VLIN Y + 9,Y + 12 AT X + 4
4682  VLIN Y + 9,Y + 12 AT X + 5
4684  VLIN Y + 9,Y + 12 AT X - 5
4690  COLOR= 0
4692  VLIN Y + 10,Y + 11 AT X - 5
4694  VLIN Y + 10,Y + 11 AT X + 5
4698  COLOR= 11
4700  VLIN Y + 14,Y + 15 AT X - 1
4702  VLIN Y + 14,Y + 15 AT X + 1
4704  HLIN X - 1,X + 1 AT Y + 19
4706  PLOT X - 2,Y + 18: PLOT X + 2,Y +
    18
4710  COLOR= 6
4712  VLIN Y + 19,Y + 24 AT X - 4
4714  VLIN Y + 19,Y + 24 AT X + 4
4716  VLIN Y + 20,Y + 24 AT X + 3
4718  VLIN Y + 20,Y + 24 AT X - 3
4720  HLIN X - 2,X + 2 AT Y + 21
4722  HLIN X - 2,X + 2 AT Y + 22
4728  VTAB 23: HTAB 20: PRINT "LAMB"
4729  RETURN
4730 XLN = 13
4732  FOR I = 1 TO 12
4734 DURV(I) = BDUR
4735  NEXT I
4736 DURV(13) = BDUR + 2 * DRINC
4737 DURV(7) = DURV(13):DURV(10) = DUR
    V(13)
4740 TN(1) = 3:TN(2) = 2:TN(3) = 1:TN(
    4) = 2
4742 TN(5) = 3:TN(6) = 3:TN(7) = 3
4744 TN(8) = 2:TN(9) = 2:TN(10) = 2
4746 TN(11) = 3:TN(12) = 5:TN(13) = 5
4748  GOSUB 82
4750  RETURN
4800  GOSUB 4810
4803  GOSUB 4920
4804  GOTO 100
4810 X =  FN RN(20) + 9
4811 Y =  FN RN(5) + 7
4812  COLOR= 10
4813  VLIN Y - 5,Y + 4 AT X - 5
4814  VLIN Y - 6,Y + 3 AT X - 4
4815  VLIN Y - 5,Y + 4 AT X + 5
4816  VLIN Y - 6,Y + 3 AT X + 4
4818  VLIN Y - 4,Y + 2 AT X - 3
4820  VLIN Y - 4,Y + 2 AT X + 3
4822  VLIN Y - 1,Y + 1 AT X - 2
4824  VLIN Y - 1,Y + 1 AT X + 2
4825  COLOR= 15
4826  FOR J = 0 TO 5
4828  HLIN X - 6 + J,X + 6 - J AT Y +
    5 - J
4830  NEXT J
4831  HLIN X - 6,X + 6 AT Y + 5
4832  HLIN X - 7,X + 7 AT Y + 6
4834  FOR J = 7 TO 11
4836  HLIN X - 4,X + 4 AT Y + J
4838  NEXT J
4840  FOR J = 12 TO 14
4842  HLIN X - 7,X + 7 AT Y + J
4844  NEXT J
4946  FOR J = 15 TO 17
4848  HLIN X - 6,X + 6 AT Y + J
4850  NEXT J
4852  HLIN X - 5,X + 5 AT Y + 18
4854  HLIN X - 5,X + 5 AT Y + 19
4856  HLIN X - 4,X + 4 AT Y + 20
4858  HLIN X - 4,X + 4 AT Y + 21
4860  COLOR= 3
4862  VLIN Y + 7,Y + 11 AT X - 7
4864  VLIN Y + 7,Y + 11 AT X + 7
4866  VLIN Y + 7,Y + 11 AT X - 6
4868  VLIN Y + 7,Y + 11 AT X + 6
4870  VLIN Y + 7,Y + 11 AT X + 5
4872  VLIN Y + 7,Y + 11 AT X - 5
4876  COLOR= 10
4878  VLIN Y + 14,Y + 20 AT X - 2
4880  VLIN Y + 14,Y + 20 AT X + 2
4882  PLOT X - 3,Y + 13: PLOT X + 3,Y +
    13
4886  COLOR= 11
4888  FOR J = 21 TO 26
4890  HLIN X - 3,X + 3 AT Y + J
4892  NEXT J
4894  VLIN Y + 22,Y + 25 AT X - 4
4896  VLIN Y + 22,Y + 25 AT X + 4
4898  HLIN X - 2,X + 2 AT Y + 27
4900  COLOR= 0
4902  VLIN Y + 22,Y + 23 AT X - 1
4904  VLIN Y + 22,Y + 23 AT X + 1
4905  COLOR= 8
4906  VLIN Y + 9,Y + 10 AT X - 7
4907  VLIN Y + 9,Y + 10 AT X + 7
4908  VLIN Y + 8,Y + 10 AT X - 6
4909  VLIN Y + 8,Y + 10 AT X + 6
4910  COLOR= 1
4912  HLIN X - 1,X + 1 AT Y + 26
4914  PLOT X - 2,Y + 25: PLOT X + 2,Y +
    25
4916  VTAB 23: HTAB 20: PRINT "HORSE"
4918  RETURN
4920  FOR I = 1 TO 10
4921 DURV(I) = BDUR
4922  NEXT I
4923 DURV(11) = BDUR + 3 * DRINC
4924 DURV(9) = DURV(11):DURV(7) = BDUR
    + 2 * DRINC
4925 TN(1) = 5:TN(2) = 5:TN(3) = 3:TN(
    4) = 5
4926 TN(5) = 6:TN(6) = 5:TN(7) = 3
4927 TN(8) = 3:TN(9) = 2
4928 TN(10) = TN(8):TN(11) = TN(9)
4929 XLN = 11
4930  GOSUB 82
4940  RETURN
5062  HLIN X - 4,X - 2 AT Y + 8: HLIN
    X + 2,X + 4 AT Y + 8
5064  HLIN X - 4,X - 2 AT Y + 9: HLIN
    X + 2,X + 4 AT Y + 9
5096  COLOR= 1: PLOT X - 4,Y + 24: PLOT
    X + 4,Y + 24
5100  GOSUB 5110
5106  GOTO 100
5110 Y =  FN RN(5) + 8
5111 X =  FN RN(15) + 10
5114  COLOR= 3
5116  HLIN X - 1,X + 1 AT Y
5118  HLIN X - 2,X + 2 AT Y + 1
5120  HLIN X - 3,X + 3 AT Y + 2
5122  FOR J = 3 TO 19
5124  HLIN X - 4,X + 4 AT Y + J
5126  NEXT J
5128  HLIN X - 3,X + 3 AT Y + 20
5130  HLIN X - 3,X + 3 AT Y + 21
5136  COLOR= 15
5138  HLIN X - 9,X - 5 AT Y + 3
5140  HLIN X - 9,X - 5 AT Y + 4
5142  HLIN X - 9,X - 5 AT Y + 5
5144  HLIN X + 5,X + 9 AT Y + 3
5146  HLIN X + 5,X + 9 AT Y + 4
```

```
5148  HLIN X + 5,X + 9 AT Y + 5
5150  HLIN X - 7,X - 5 AT Y + 6
5152  HLIN X + 5,X + 7 AT Y + 6
5156  FOR I = 1 TO 2
5158  VLIN Y + 7,Y + 10 AT X - I
5160  VLIN Y + 7,Y + 10 AT X + I
5162  NEXT I
5166  FOR I = 0 TO 2
5168  VLIN Y + 22,Y + 25 - I AT X + I
5170  VLIN Y + 22,Y + 25 - I AT X - I
5172  NEXT I
5176  COLOR= 10
5178  FOR I = 16 TO 20
5180  HLIN X - 2,X + 2 AT Y + I
5182  NEXT I
5186  COLOR= 1
5188  HLIN X - 1,X + 1 AT Y + 20
5190  PLOT X - 2,Y + 19: PLOT X + 2,Y +
    19
5194  COLOR= 0
5196  PLOT X - 2,Y + 8: PLOT X + 2,Y +
    8
5198  PLOT X - 1,Y + 17: PLOT X + 1,Y +
    17
5202  COLOR= 13
5204  VLIN Y - 7,Y + 2 AT X - 3
5206  VLIN Y - 7,Y + 2 AT X + 3
5210  VTAB 23: HTAB 20: PRINT "GOAT"
5220  RETURN
5250  REM   - I HERE
5252  GR : CALL  - 936
5255 SD = 7:S1 = SD
5256 X = 20:Y = 22
5260  COLOR= 13
5262  FOR I = 0 TO 1
5264  HLIN X - SD + 1,X + SD - 1 AT Y +
    I
5266  NEXT I
5268  FOR J = 2 TO 14 STEP 3
5270 SD = SD - 1
5272  FOR K = 0 TO 2
5274  HLIN X - SD,X + SD AT Y + J + K
5276  NEXT K
5278  NEXT J
5282 R =  FN RN(5) + 1
5284  IF R > 2 THEN C = 15
5286  IF R = 1 THEN C = 8
5288  IF R = 2 THEN C = 11
5290  COLOR= C
5294  FOR I = 1 TO 8
5295  HLIN X - S1,X + S1 AT Y - I
5296  NEXT I
5298  FOR I = 9 TO 13
5300  HLIN X - S1 + 1,X + S1 - 1 AT Y -
    I
5302  NEXT I
5304  HLIN X - S1 / 2,X + S1 / 2 AT Y -
    14
5306  HLIN X - S1 / 2,X + S1 / 2 AT Y -
    15
5310  IF R > 1 THEN 5315
5312  VTAB 22: HTAB 12: PRINT "CHOCOLA
    TE"
5314  GOTO 5350
5315  IF R > 2 THEN 5320
5317  VTAB 22: HTAB 12: PRINT "STRAWBE
    RRY"
5318 C1 = 1
5319  GOTO 5340
5320  IF R > 3 THEN 5325
5322  VTAB 22: HTAB 12: PRINT "CHERRY"
5323 C1 = 1
5324  GOTO 5340
5325  IF R > 4 THEN 5330
5327  VTAB 22: HTAB 12: PRINT "CHOCOLA
    TE CHIP"
5328 C1 = 8
5329  GOTO 5340
5330  VTAB 22: HTAB 12: PRINT "VANILLA
    "
5332  GOTO 5350
5340 XR =  FN RN(15) + 4
5342  COLOR= C1
5343  FOR I = 1 TO XR
5344 XP = X - S1 + 1 +  FN RN(2 * S1)
5345 YP = Y -  FN RN(13) - 1
5346  PLOT XP,YP
5347  NEXT I
5350  VTAB 23: HTAB 12: PRINT "ICE CRE
    AM CONE"
5352  GOTO 100
5360  REM   - QUICK
5362 WT = 10
5364  VTAB 22: HTAB 15: PRINT "QUICK"
5366  FOR K = 1 TO 50: NEXT K
5368  GOSUB 5504
5370  FOR K = 1 TO 100: NEXT K
5372 WT = 150
5374  VTAB 22: HTAB 15: PRINT "SLOW  "
5376  GOSUB 5504
5378  GOTO 100
5400 OO = 1
5401  GR
5402  GOSUB 8100
5404 X = 5:Y = 15: GOSUB 6350
5406 X = X + 6: GOSUB 6230
5408 X = X + 5: GOSUB 6330
5410 X = X + 5: GOSUB 6200
5412 X = X + 5: GOSUB 6200
5414  GOTO 100
5500 WT = 100: GOSUB 5504
5503  GOTO 100
5504  GR
5505  VTAB 23: HTAB 15: PRINT "WORM"
5506 XH = 15
5508 Y = 25
5509  GOTO 5513
5510  COLOR= 0
5511  FOR I = 0 TO 5: HLIN XH - 15,XH AT
    Y - I
5512  NEXT I
5513  COLOR= 12
5514  FOR J = 1 TO 3
5516  HLIN XH - 15,XH AT Y + J
5518  NEXT J
5520  COLOR= 1
5522  PLOT XH - 1,Y + 2: PLOT XH + 1,Y
5525  FOR I = 1 TO WT: NEXT I
5530  COLOR= 12
5532 XM = XH - 4:XL = XH - 6
5536  FOR J = 1 TO 3
5538  HLIN XL - 2,XM + 2 AT Y - 1 - J
5540  NEXT J
5542  VLIN Y - 2,Y AT XL - 1
5544  VLIN Y - 2,Y AT XL - 2
5546  VLIN Y - 2,Y AT XM + 1
5548  VLIN Y - 2,Y AT XM + 2
5550  COLOR= 0
5552  FOR I = 1 TO 3
5554  HLIN XL,XM AT Y + I
5555  HLIN XH - 15,XH - 10 AT Y + I
5556  NEXT I
5560  FOR K = 1 TO WT: NEXT K
5562 XH = XH + 5
5564  IF XH < 39 THEN 5510
5570  RETURN
5600  COLOR= 1
5606  FOR R = 18 TO 39
5608  HLIN 1,23 AT R
5610  NEXT R
5614  COLOR= 8
5616  FOR R = 30 TO 39
5618  HLIN 6,15 AT R
5620  NEXT R
5630  COLOR= 5
5632  FOR R = 11 TO 36
5634  HLIN 24,37 AT R
5636  NEXT R
5640  COLOR= 15
5642  FOR R = 5 TO 10
5644  HLIN 24,37 AT R
5645  NEXT R
5646  FOR R = 2 TO 4
5647  HLIN 26,35 AT R
5648  NEXT R
5650  HLIN 28,33 AT 0
5651  HLIN 28,33 AT 1
5660  COLOR= 8
5662  FOR R = 5 TO 17
5664  HLIN 0,24 AT R
5666  NEXT R
5680  COLOR= 13
5682  FOR I = 1 TO 10
5684  PLOT 5 + I,29 + I
5685  PLOT 16 - I,29 + I
5686  NEXT I
5699  RETURN
5700  COLOR= 2
5709 PL = 14
5710  FOR J = 1 TO 7
5712  IF J = 1 OR J = 4 THEN NS = 2
5713  IF J <  > 1 AND J <  > 4 THEN NS
    = 4
5714  FOR K = 1 TO NS
5715  HLIN 0,4 * J - 1 AT PL + K
5716  NEXT K
5717 PL = PL + NS
5718  NEXT J
5722  COLOR= 13
5723  VTAB 23: HTAB 20: PRINT "UP"
5724  FOR W = 1 TO 200: NEXT W
5725 X = 29:Y = 38
5726  PLOT X,Y:TN(1) = 1:DURV(1) = BDU
    R - DRINC
5727  GOSUB 5741
5728  FOR X = 26 TO 2 STEP  - 4
5729 Y = Y - 4
5730  IF X = 14 OR X = 2 THEN Y = Y +
    2
5732  PLOT X,Y
5733  GOSUB 5740
5734  NEXT X
5735 X = 2: PLOT X,Y
5736  GOTO 5750
5740 TN(1) =  INT ((34 - X) / 4):DURV(
    1) = BDUR - DRINC
5741 XLM = 1: GOSUB 82
5742  FOR W = 1 TO 100: NEXT W
5744  COLOR= 0: PLOT X,Y
5746  COLOR= 13
```

```
5748  RETURN
5750  COLOR= 9
5752  VTAB 23: HTAB 20: PRINT "DOWN"
5754  FOR W = 1 TO 200: NEXT W
5756  FOR X = 2 TO 26 STEP 4
5758  PLOT X,Y
5760  GOSUB 5740
5762 Y = Y + 4
5763  IF X = 14 OR X = 2 THEN Y = Y -
     2
5764  NEXT X
5766  PLOT 29,38
5768 TN(1) = 1:DURV(1) = BDUR - DRINC
5769 XLN = 1: GOSUB 82
5770  GOTO 100
5800  REM  - KOALA
5804 X =  FN RN(3) + 17:Y =  FN RN(8) +
     5
5806  GR : CALL  - 936
5810  COLOR= 8
5812  FOR I = 0 TO 5
5814  HLIN X - 3 - I,X + 3 + I AT Y +
     I
5816  NEXT I
5818  HLIN X - 8,X + 8 AT Y + 6
5820  FOR J = 7 TO 16
5822  HLIN X - 9,X + 9 AT Y + J
5824  NEXT J
5832  FOR I = 0 TO 3
5834  HLIN X - 9 + I,X + 9 - I AT Y +
     17 + I
5836  NEXT I
5838  HLIN X - 3,X + 3 AT Y + 21
5844  COLOR= 15
5848  HLIN X - 13,X - 6 AT Y - 3
5850  HLIN X + 6,X + 13 AT Y - 3
5852  HLIN X - 13,X - 6 AT Y - 2
5854  HLIN X + 6,X + 13 AT Y - 2
5856  HLIN X - 15,X - 4 AT Y - 1
5858  HLIN X - 15,X - 4 AT Y
5860  HLIN X + 4,X + 15 AT Y - 1
5862  HLIN X + 4,X + 15 AT Y
5864  HLIN X - 15,X - 5 AT Y + 1
5866  HLIN X + 5,X + 15 AT Y + 1
5868  FOR I = 2 TO 5
5870  HLIN X - 16,X - 4 - I AT Y + I
5872  HLIN X + 4 + I,X + 16 AT Y + I
5874  NEXT I
5876  HLIN X - 16,X - 9 AT Y + 6
5878  HLIN X + 9,X + 16 AT Y + 6
5880  FOR J = 7 TO 11
5882  HLIN X - 16,X - 10 AT Y + J
5884  HLIN X + 10,X + 16 AT Y + J
5886  NEXT J
5888  HLIN X - 15,X - 10 AT Y + 12
5890  HLIN X + 10,X + 15 AT Y + 12
5892  HLIN X - 14,X - 10 AT Y + 13
5894  HLIN X + 10,X + 14 AT Y + 13
5896  VLIN Y + 7,Y + 10 AT X - 3
5898  VLIN Y + 7,Y + 10 AT X + 3
5900  VLIN Y + 7,Y + 10 AT X - 4
5902  VLIN Y + 8,Y + 10 AT X - 5
5904  VLIN Y + 7,Y + 10 AT X + 4
5906  VLIN Y + 8,Y + 10 AT X + 5
5910  COLOR= 0
5912  HLIN X - 1,X + 1 AT Y + 11
5914  HLIN X - 2,X + 2 AT Y + 12
5916  HLIN X - 2,X + 2 AT Y + 13
5918  HLIN X - 3,X + 3 AT Y + 14
5920  HLIN X - 3,X + 3 AT Y + 15
5922  HLIN X - 3,X + 3 AT Y + 16
5923  HLIN X - 3,X + 3 AT Y + 17
5924  VLIN Y + 8,Y + 9 AT X - 4
5926  VLIN Y + 8,Y + 9 AT X + 4
5932  HLIN X - 1,X + 1 AT Y + 20
5934  PLOT X - 2,Y + 19: PLOT X + 2,Y +
     19
5940  VTAB 23: HTAB 12: PRINT "KIRI'S
     KOALA WALLY"
5950 XLN = 20
5954  FOR I = 1 TO 20:TN(I) = 5: NEXT
     I
5956 TN(5) = 3:TN(6) = 8:TN(7) = 8:TN(
     8) = 8
5958 TN(9) = 7:TN(10) = 6:TN(14) = 6
5960 TN(18) = 4:TN(19) = 3:TN(20) = 2
5962  FOR I = 1 TO 20:DURV(I) = BDUR: NEXT
     I
5964 HF = BDUR - DRINC:DT = BDUR + DRI
     NC
5966 DURV(2) = 3 * BDUR / 4:DURV(3) =
     BDUR / 4
5967 DURV(7) = 3 * BDUR / 4:DURV(8) =
     BDUR / 4
5968 DURV(12) = 3 * BDUR / 4:DURV(13) =
     BDUR / 4
5970 DURV(15) = 3 * BDUR / 4:DURV(16) =
     BDUR / 4
5972 DURV(18) = 3 * BDUR / 4:DURV(19) =
     BDUR / 4
5974 DURV(20) = BDUR + 2 * DRINC
5976  GOSUB 82
5980  GOTO 100
6000 OO = 1
6160  REM  - A
6162  VLIN Y,Y + 4 AT X
6163  VLIN Y,Y + 4 AT X + 3
6165  HLIN X,X + 3 AT Y
6167  HLIN X,X + 3 AT Y + 2
6169  RETURN
6170  REM  - B
6172  VLIN Y,Y + 4 AT X
6174  PLOT X + 3,Y + 1: PLOT X + 3,Y +
     3
6176  HLIN X,X + 3 AT Y
6177  HLIN X,X + 3 AT Y + 2
6178  HLIN X,X + 3 AT Y + 4
6179  RETURN
6180  REM  - C
6182  HLIN X,X + 3 AT Y
6184  HLIN X,X + 3 AT Y + 4
6186  VLIN Y,Y + 4 AT X
6189  RETURN
6190  REM  - D
6192  VLIN Y,Y + 4 AT X
6193  VLIN Y + 1,Y + 3 AT X + 3
6195  HLIN X,X + 2 AT Y
6197  HLIN X,X + 2 AT Y + 4
6199  RETURN
6200  REM  - E
6202  VLIN Y,Y + 4 AT X
6204  HLIN X,X + 3 AT Y
6206  HLIN X,X + 2 AT Y + 2
6208  HLIN X,X + 3 AT Y + 4
6209  RETURN
6210  REM  - F
6212  VLIN Y,Y + 4 AT X
6214  HLIN X,X + 3 AT Y
6216  HLIN X,X + 2 AT Y + 2
6218  RETURN
6220  REM  - G
6222  VLIN Y,Y + 4 AT X
6224  HLIN X,X + 3 AT Y
6226  HLIN X,X + 3 AT Y + 4
6227  HLIN X + 2,X + 3 AT Y + 2
6228  VLIN Y + 2,Y + 4 AT X + 3
6229  RETURN
6230  REM  - H
6232  VLIN Y,Y + 4 AT X
6234  VLIN Y,Y + 4 AT X + 3
6236  HLIN X,X + 3 AT Y + 2
6238  RETURN
6240  REM  - I
6242  VLIN Y,Y + 4 AT X + 2
6244  HLIN X + 1,X + 3 AT Y
6246  HLIN X + 1,X + 3 AT Y + 4
6248  RETURN
6250  REM  - J
6252  HLIN X,X + 3 AT Y + 4
6254  VLIN Y,Y + 4 AT X + 3
6256  VLIN Y + 3,Y + 4 AT X
6258  RETURN
6260  REM  - K
6262  VLIN Y,Y + 4 AT X
6264  PLOT X + 1,Y + 2
6266  PLOT X + 2,Y + 1: PLOT X + 2,Y +
     3
6268  PLOT X + 3,Y: PLOT X + 3,Y + 4
6269  RETURN
6270  REM  - L
6272  VLIN Y,Y + 4 AT X
6274  HLIN X,X + 3 AT Y + 4
6279  RETURN
6280  REM  - M
6282  VLIN Y,Y + 4 AT X
6284  VLIN Y,Y + 4 AT X + 4
6285  VLIN Y,Y + 1 AT X + 1
6286  VLIN Y,Y + 1 AT X + 3
6287  PLOT X + 2,Y + 2
6288  RETURN
6290  REM  - N
6292  VLIN Y,Y + 4 AT X
6294  VLIN Y,Y + 4 AT X + 4
6295  PLOT X + 1,Y: PLOT X + 3,Y + 4
6296  FOR IJ = 1 TO 3
6297  PLOT X + IJ,Y + IJ
6298  NEXT IJ
6299  RETURN
6300  REM  - O
6302  VLIN Y,Y + 4 AT X
6304  VLIN Y,Y + 4 AT X + 3
6306  HLIN X,X + 3 AT Y
6308  HLIN X,X + 3 AT Y + 4
6309  RETURN
6310  REM  - P
6312  VLIN Y,Y + 4 AT X
6314  VLIN Y,Y + 2 AT X + 3
6315  HLIN X,X + 3 AT Y
6317  HLIN X,X + 2 AT Y + 2
6319  RETURN
6320  REM  - Q
6322  HLIN X,X + 3 AT Y
6323  HLIN X,X + 4 AT Y + 4
6325  VLIN Y,Y + 4 AT X
6326  VLIN Y,Y + 4 AT X + 3
6327  PLOT X + 2,Y + 3
6329  RETURN
6330  REM  - R
6332  GOSUB 6312
6334  PLOT X + 2,Y + 3: PLOT X + 3,Y +
     4
6336  PLOT X + 1,Y
6339  RETURN
```

```
6340  REM   - S
6342  HLIN X,X + 3 AT Y
6344  HLIN X,X + 3 AT Y + 2
6345  HLIN X,X + 3 AT Y + 4
6346  VLIN Y,Y + 2 AT X
6348  VLIN Y + 2,Y + 4 AT X + 3
6349  RETURN
6350  REM   - T
6352  HLIN X,X + 4 AT Y
6354  VLIN Y,Y + 4 AT X + 2
6359  RETURN
6360  REM   - U
6362  VLIN Y,Y + 4 AT X
6364  VLIN Y,Y + 4 AT X + 3
6366  HLIN X,X + 3 AT Y + 4
6369  RETURN
6370  REM   - V
6372  VLIN Y,Y + 2 AT X
6374  VLIN Y,Y + 2 AT X + 4
6375  PLOT X + 1,Y + 3: PLOT X + 3,Y +
    3
6377  PLOT X + 2,Y + 4
6379  RETURN
6380  REM   - W
6382  VLIN Y,Y + 4 AT X
6383  VLIN Y,Y + 4 AT X + 4
6385  VLIN Y + 3,Y + 4 AT X + 1: VLIN
    Y + 3,Y + 4 AT X + 3
6387  PLOT X + 2,Y + 2
6389  RETURN
6390  REM   - X
6392  FOR I = 0 TO 4
6394  PLOT X + I,Y + I
6395  PLOT X + 4 - I,Y + I
6396  NEXT I
6399  RETURN
6400  REM   - Y
6402  VLIN Y + 2,Y + 4 AT X + 2
6404  PLOT X,Y: PLOT X + 1,Y + 1
6406  PLOT X + 4,Y: PLOT X + 3,Y + 1
6409  RETURN
6410  REM   - Z
6412  HLIN X,X + 4 AT Y
6413  HLIN X,X + 4 AT Y + 4
6415  HLIN X + 3,X + 4 AT Y + 1
6416  HLIN X,X + 1 AT Y + 3
6417  PLOT X + 2,Y + 2
6419  RETURN
6450  REM   - INI-
6460  CALL  - 936: GR
6462  GOSUB 6530
6464 X = 1:Y = 1
6465  FOR Q = 0 TO 240 STEP 10
6466  COLOR=  FN RN(15) + 1
6470  ON  INT (Q / 10) + 1 GOSUB 6160,
    6170,6180,6190,6200,6210,6220,6230
    ,6240,6250,6260,6270,6280,6290,630
    0,6310,6320,6330,6340,6350,6360,63
    70,6380,6390,6400
6478 X = X + 7
6480  IF X <  = 34 THEN 6490
6485 Y = Y + 6
6486 X = 1
6490  NEXT Q
6491 X = 15
6492  GOSUB 6410: FOR JT = 1 TO 100: NEXT
    JT
6493  COLOR= 15:X = 3:Y = 6:XLN = 1
6494  FOR JT = 1 TO 29
6495  PLOT X,Y
6496 TN(1) = TN(JT):DURV(1) = DURV(JT)
6497  GOSUB 82
6498  COLOR= 0: PLOT X,Y: COLOR= 15:X =
    X + 7
6499  IF JT = 19 THEN X = X - 7
6500  IF JT = 24 OR JT = 25 OR JT = 26
    THEN X = X - 7
6502  IF X < 32 THEN 6504
6503 X = 3:Y = Y + 6
6504  NEXT JT
6505 X = 17:Y = 36: PLOT X,Y
6506 TN(1) = TN(30):DURV(1) = DURV(30)
    : GOSUB 82
6507  COLOR= 0: PLOT X,Y: COLOR= 15: PLOT
    17,38
6508  GOSUB 6695
6509  GOTO 100
6510 XLN = 2:TN(2) = 4:DURV(2) = BDUR
6512  GOSUB 82
6513 XLN = 1
6514  GOTO 6478
6515 XLN = 4:TN(2) = 5:TN(3) = 4:TN(4)
    = 4
6516 D2 = BDUR - DRINC
6517 DURV(2) = D2:DURV(3) = D2:DURV(4)
    = D2
6518  GOSUB 82
6519 XLN = 1: GOTO 6478
6520  GOTO 100
6530  FOR I = 1 TO 26:TN(I) = 2: NEXT
    I
6535 TN(1) = 1:TN(2) = 1:TN(3) = 5:TN(
    4) = 5
6537 TN(5) = 6:TN(6) = 6:TN(7) = 5:TN(
    8) = 4
6540 TN(9) = 4:TN(10) = 3:TN(11) = 3
6542 TN(16) = 1:TN(17) = 5:TN(18) = 5:
    TN(19) = 4
6545 TN(20) = 4:TN(21) = 3:TN(22) = 3
6547 TN(24) = 5:TN(25) = 5:TN(26) = 4:
    TN(27) = 4
6550 TN(28) = 3:TN(29) = 3:TN(30) = 2
6555  FOR I = 1 TO 30:DURV(I) = BDUR: NEXT
    I
6560 DURV(7) = 2 * BDUR
6562 N1 = BDUR - DRINC:N2 = BDUR + DRI
    NC
6563 N3 =  INT (BDUR / 2)
6565 DURV(12) = N3:DURV(13) = N3:DURV(
    14) = N3:DURV(15) = N3
6567 DURV(16) = 2 * BDUR
6570 DURV(23) = 2 * BDUR:DURV(30) = 2 *
    BDUR
6575 XLN = 30
6585  RETURN
6600  GR : CALL  - 936
6610  GOSUB 6530
6615  COLOR= 4
6635 XLN = 16: GOSUB 82
6640 X = 1:Y = 15
6642  GOSUB 6180
6644 X = X + 5: GOSUB 6300
6646 X = X + 5: GOSUB 6280
6648 X = X + 6: GOSUB 6310
6650 X = X + 5: GOSUB 6360
6652 X = X + 5: HLIN X,X + 2 AT Y
    + 2
6656 X = 8:Y = 22
6660  FOR J = 17 TO 23
6664 TN(J - 16) = TN(J)
6666 DURV(J - 16) = DURV(J)
6667  NEXT J
6668 XLN = 7
6669  GOSUB 82
6670 X = 8:Y = 22
6672  GOSUB 6230
6674 X = X + 5: GOSUB 6300
6676 X = X + 5: GOSUB 6360
6678 X = X + 5: GOSUB 6340
6680 X = X + 5: GOSUB 6200
6684  FOR J = 24 TO 30
6686 TN(J - 23) = TN(J)
6688 DURV(J - 23) = DURV(J)
6689  NEXT J
6690 XLN = 7
6691  GOSUB 82
6692  GOSUB 6695
6693  GOTO 6704
6695  GOSUB 6530
6696 DURV(12) = BDUR:DURV(13) = B
    DUR
6697 TN(14) = 1:DURV(14) = 2 * BD
    UR
6698 XLN = 14
6699  GOSUB 82
6700  RETURN
6704  CALL  - 936
6706  PRINT : FOR I = 1 TO 300: NEXT

6708  PRINT "HIT ANY KEY TO CONTI
    NUE": GR
6710  GOTO 100
8000 PP = P - 159
8010  ON PP GOTO 200,250,300,350,
    400,450,500,550,600,650,700,
    750,800
8015 PP = PP - 13
8020  ON PP GOTO 850,900,950,1000
    ,1050,1100,1150,1200,1250,13
    00,1350,1400
8025 PP = PP - 12
8030  ON PP GOTO 1450,1500,1550,1
    600,1650,1700,1750,1800,1850
    ,1900,1950,2000
8035 PP = PP - 12
8040  ON PP GOTO 2050,2100,2150,2
    200,2250,2300,2350,2400,2450
    ,2500
8045 PP = PP - 10
8050  ON PP GOTO 2550,2600,2650,2
    700,2750,2800,2850,2900,2950
    ,3000,3050,3100
8060  GOTO 850
8095  FOR KX = 1 TO JL
8096  IF  INT (KX / 2) * 2 = KX THEN
    COLOR= 13
8097  IF  INT (KX / 2) * 2 <  > K
    X THEN  COLOR= 4
8098  GOTO 2317
8100  COLOR=  FN RN(15) + 1
8110  RETURN
8120  IF  INT (NX / 2) * 2 = NX THEN
    COLOR= 6
8124  IF  INT (NX / 2) * 2 <  > N
    X THEN  COLOR= 15
8126  GOTO 2327
8130  IF  INT (NX / 2) * 2 = NX THEN
    COLOR= 1
8134  IF  INT (NX / 2) * 2 <  > N
    X THEN  COLOR= 8
8138  GOTO 2334
9999  END
```

特惠訂閱

最後特惠機會

優待至三月十五日止日

由即日開始訂閱電腦時代一年（十二期），可獲贈由 CompuHouse 製作之 CompuHouse Disk* 一隻，只需付出港幣九十六元正（包括郵費）

同時訂購電腦時代＋CompuHouse 磁碟一年（十二期），只需付出港幣三百五十四元正（連郵費）

請填妥下列表格連同劃線支票寄回：

九龍中央郵政信箱 71193 號
電腦時代出版社收

*本刊每期刊出之所有程式，均由 CompuHouse 全部打入一張磁碟，省卻讀者自行打入程式的時間及可免卻捉錯打字的煩悶步驟。CompuHouse 製作之磁碟是採用高級貨，保證品質優越。

如要掛號，十二期加付港幣四十八元

（請用英文填寫，方便電腦處理）　（只適用於本港訂戶）　信封面請註明「訂閱」

本人欲訂閱

□電腦時代一年（十二期）HK$96
□電腦時代＋CompuHouse磁碟一年（十二期）HK$354

由第___期開始

姓名：________________

地址：________________

電話：________________

附上劃線支票 No. ________________

支票收款人請填寫 "Computing Age Publisher"

請寄九龍中央郵箱71193號

NO. __________



★ 學習尖端科技

★ 邁向成功坦途

現代電腦系統訓練中心

MODERN COMPUTER SYSTEM TRAINING CENTRE

每月均開新班

（軟件及硬件課程）

（1）Apple II 及其他小型電腦修理班（全期6個月，80小時課程）
已開辦半年，以造就電腦維修人才爲目的，隔月開新班 $1500.00

（2）電腦磁碟開鎖及加鎖學習班（18小時課程） $550.00
每月均開新班

（3）CP/M入門（全期20小時） $300.00

（4） BASE-II 精修班（全期20小時） $300.00

創新課程：

Programmer Training Course（全期80小時，共6個月）

A班：入學資格——中五程度或以上及有BASIC Programming知識 $1200.00

B班：入學資格——中五程度或以上，無需任何電腦知識 $1500.00

（每月只收壹班，每班只收10人）

本課程專爲中學或以上人士提供全面性電腦操作及軟件程序設計之訓練。內容教學與實習並重，當課程進行至兩個月左右，即分配學員做Project, 分組完成，由資深程式設計師指導，並將合組公司，將所完成之Programmes申請專利或賣與有需要之機構等，而所得利潤，學員亦可分享，從而達到成爲眞正Programmer之目的；並非一般只偏重灌輸生硬電腦智識之電腦課程可比。

——以上課程3人以上報名尚有9折優待——

創辦兩年　電腦碩士教任

本校師資
(1)香港及中文大學電腦系，英美加電腦系畢業生。
(2)英國曼徹斯特大學電腦碩士，英國電腦學會會員。

本中心開放時間：10：00A.M—10：00P.M　電話：3-601531

地址：九龍深水埗福華街156號黃金商場地庫20號室

分址：荃灣大壩街67號閣樓　0-429147　（精通電腦中心）

旺角太子道141號長榮商業大厦303A號（彌敦道口）3-869085

LEE'S ELECTRONIC CO. 李氏電子公司

灣仔總店　北角分店

大空型 JOYSTICK

打破傳統設計將兩個發射掣安裝在控制桿上，更易操作，可配合最新推出JOYSTICK CARD使用，0-255讀數可停留在任何位置，AUTO FIRE,電子控制，可以適應任何新舊GAME使用非一般 VR式可比。

AUTO FIRE掣



ZETA SLIM COMPUTER DATA RECORDER

電腦專用錄音機
獨有SAVE及LOAD鑑聽和指示
AUTO STOP, 帶長COUNTER
FOR APPLE COMPUTER
和多種電腦使用，比其他產品更易LOAD帶和SAVE



"香港區總代理，歡迎批發"

SPEECH CARD II

中文，英文說話卡，更可幫你讀 CATALOG, DOS COMMAND, 外接式高品質小型喇叭箱，聲音響亮清晰 Software + Manual

只售壹佰餘圓

全新推出 JOY-PORT III

只售壹佰餘圓

特別設計全新綫路，絕對可玩全部 GAME
可以選擇 AUTO FIRE 配合 QUICK SHOT JOYSTICK

注意 QUICK SHOT JOYSTICK, 如果唔加 JOYPORT CARD, 唔可以玩 LADY TUT，或者將來新 GAME。

營業時間：上午11:30 AM至下午9:30PM

香港北角英皇道255號國都商場3樓213室
電話：5-718392
Shop No. 213 Olympia Shopping Centre,
2/F., 255 King's Rd., North Point, H.K.
TEL: 5-718392

香港灣仔道128-150號海運商場1　1-2室
電話：5-8921792
Shop No. 1-2 Ocean Shopping Centre,
1/F., 128-150 Wanchai Rd., Hong Kong.
TEL: 5-8921792

COMPUTING AGE

電腦時代

第四期
一九八四年二月二十日出版
本刊每月二十五日出版
每本訂價港幣八元正

第4期程式磁碟同日出版
各特約經銷處有售　地址請參閱封底廣告

出　　版/ 電腦時代出版社
地　　址/ 英皇道 121 號嘉信大廈廿九樓A 4
編　　輯/ 電腦時代編輯部
通 訊 地 址/ 九龍中央郵箱71193號
通 訊 電 話/ 3-7776383
軟 件 支 援/ CompuHouse
讀者服務部/ 深水埗南昌街白田購物中心二樓 197 室
承　　印/ 永恒印刷公司
葵涌工業街美安工業大廈12字樓 A座
發　　行/ 同德書局
九龍西洋菜街 193 號

今期內容

■ Program saved on disk